GERENCIAMENTO DE REDES DE COMPUTADORES COM O USO DO RACIOCÍNIO BASEADO EM CASOS E FERRAMENTAS AUXILIARES

Frederico Sauer Guimarães Oliveira

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

_____
Prof. Nelson Francisco Favilla Ebecken, D.Sc

_____
Prof. Alexandre Gonçalves Evsukoff, Dr.

_____
Prof. Aloysio de Castro Pinto Pedroza, Dr.

_____
Prof. Beatriz de Souza Leite Pires de Lima, D.Sc.

_____
Prof. Ronaldo Moreira Salles, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2007

OLIVEIRA, FREDERICO SAUER GUIMARÃES

Gerenciamento de Redes de Computadores com o Uso do Raciocínio Baseado em Casos e Ferramentas Auxiliares. [Rio de Janeiro] 2007

VII, 148p, 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia Civil, 2007)

Tese – Universidade Federal do Rio de Janeiro, COPPE

1. Gerenciamento de Redes
2. Raciocínio Baseado em Casos

I. COPPE/UFRJ II. Título (série)

Dedicatória

Aos meus filhos Bruno Sauer e Frederico Sauer. Que esta conquista sirva como exemplo: o universo conspira para realizarmos os nossos desejos mais íntimos. Desejem, sonhem, ousem, e o aparentemente impossível acontecerá.

A minha esposa Adriana, que sempre me apoiou, apesar de não compreender muito bem porque abdiquei dos poucos espaços de possível tranqüilidade para me empenhar neste projeto. Agora posso te levar para dançar. Você merece.

Agradecimentos

A Deus, arquiteto do universo;

A minha mãe Ione, pelo amor discreto e o cuidado indiscreto com minha educação;

Ao meu pai Francisco, pelo exemplo e pelo apoio incondicional, em todos os momentos de minha vida;

Ao professor Nelson Ebecken, cuja simplicidade e postura ocultam um dos mais completos líderes que já conheci;

Ao professor e amigo Alexandre Evsukoff, elemento-chave nesta conspiração do universo. Sem o seu apoio e confiança, jamais estaria escrevendo estas linhas;

Aos digníssimos professores Aloysio, Beatriz e Ronaldo, pelo aceite da participação na minha defesa. Todos os comentários e críticas muito me honrarão.

Aos meus melhores amigos, Arantes, Lemos e Monnerat, homens brilhantes que sempre se abstraíram de meus defeitos e limitações, enxergando apenas o pouco de bom que há em mim;

Aos meus alunos e profissionais subordinados, principais motivadores da minha busca constante pelo aprimoramento;

Aos meus Chefes Navais, especialmente os Comandantes Jacoby, Destri e Souto, que acreditaram em mim;

A Marinha, a melhor das casas, que acolhe, instrui, oferece oportunidades e justiça a que a merece;

A todos aqueles que dificultaram meu trabalho. Os obstáculos serviram para me fortalecer e acreditar que seria possível;

Aos funcionários e amigos da COPPE, especialmente o Jairo, o Telmo, e o pessoal da biblioteca do NCE;

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

GERENCIAMENTO DE REDES DE COMPUTADORES COM O USO DO RACIOCÍNIO BASEADO EM CASOS E FERRAMENTAS AUXILIARES.

Frederico Sauer Guimarães Oliveira

Setembro/2007

Orientadores: Nelson Francisco Favilla Ebecken
Alexandre Gonçalves Evsukoff

Programa: Engenharia Civil

Este trabalho propõe uma estratégia para o Gerenciamento de Redes de Computadores, combinando o Raciocínio Baseado em Casos (CBR) e ferramentas auxiliares. Estas ferramentas foram desenvolvidas para possibilitar o enriquecimento da descrição da percepção do usuário quanto à falha, com o objetivo de aumentar a similaridade com uma falha típica ou uma falha já ocorrida na rede em análise. A disponibilidade de um *test-bed* real aproxima o modelo proposto das condições da maioria das redes. Para viabilizar esta contribuição, foi implementado um sistema CBR combinando experiência especialista e dados reais. Os resultados obtidos evidenciaram o aumento da eficiência e da eficácia no gerenciamento da rede.

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

COMPUTER NETWORK MANAGEMENT WITH THE USE OF CASE-BASED REASONING AND AUXILIARY TOOLS.

Frederico Sauer Guimarães Oliveira

September/2007

Advisors: Nelson Francisco Favilla Ebecken
Alexandre Gonçalves Evsukoff

Department: Civil Engineering

This work proposes a strategy for the Computer Network Management, combining the Case-Based Reasoning (CBR) with auxiliary tools. Those tools were built to improve the user perception of the fail description, with the objective of maximizing the similarity with a typical fail or a past occurred fail. A useful test-bed binds the proposed model with the majority of Computer Networks environments. To make this contribution, a CBR system was implemented, combining specialist knowledge and real data. The obtained results showed an improved efficiency and efficacy on the network management.

Sumário

1 Introdução

Este trabalho se concentra em um dos maiores desafios impostos pelo fenômeno da globalização: a Gerência de Falhas em Redes de Computadores. A Tecnologia da Informação, através das Redes de Comunicação de Dados, permitiu que eventos ocorridos em qualquer parte do globo terrestre pudessem ser do conhecimento de qualquer habitante do planeta, desde que o mesmo possua acesso à maior Rede de Comunicação do mundo, a Internet. Com o crescimento exponencial do número de usuários da Internet, diversos problemas podem provocar falhas que degradam a performance das redes de acesso aos *backbones*, causando sensação de lentidão, intermitência ou, em seu caso mais extremo, indisponibilidade. Os Sistemas Operacionais, por sua vez, têm investido grandes esforços na criação de interfaces homem-máquina cada vez mais amigáveis[1], tornando abstrato para o usuário o que ocorre na conectividade entre o dispositivo usado para acessar a informação e o servidor onde a mesma se encontra armazenada. Nesta área de *expertise*, as redes locais interligadas através da *internet* possuem características próprias. As falhas são repetitivas e típicas, mas a indisponibilidade de sintomas claros e determinantes das respectivas causas dificulta a tarefa de manter a rede operacional com um nível de qualidade de serviço satisfatório. Em geral, as redes são administradas por técnicos com profundo conhecimento das características não só da rede, mas também do comportamento dos seus usuários, e este conhecimento especialista não é compartilhado.

Algumas propostas para minimizar estes problemas podem ser observadas na literatura, conforme apresentado na seção 1.3. Dentre estas propostas, o Raciocínio Baseado em Casos (CBR – *Case-Based Reasoning*) tem obtido resultados interessantes, por permitir o aprimoramento de algumas atividades. Um exemplo é o relacionamento com clientes em várias áreas bastante distintas, que permite a sua avaliação como uma ferramenta para a construção de "aplicações inteligentes". Uma aplicação baseada em casos provê, principalmente, uma base onde o conhecimento adquirido em experiências passadas pode ser acumulado e, com esta base, o usuário pode recuperar e reutilizar o conhecimento em uma situação não necessariamente idêntica, mas com um determinado grau de similaridade. Em alguns casos, existe a possibilidade não apenas da adaptação

[1] Nota do autor: o investimento em interfaces *user-friendly* torna a conectividade cada vez menos *network-friendly*. Poucos usuários conhecem sequer a sua arquitetura de acesso ao provedor.

de um caso semelhante à situação consultada, mas também a agregação de aspectos de difícil visibilidade para o usuário do sistema numa abordagem inicial, fazendo o que se chama de "completar" um caso. Isso possibilita a construção de uma memória corporativa, através da captura de problemas e suas respectivas soluções. Esta base de conhecimento pode então ser disponibilizada para qualquer usuário, aumentando a produtividade em várias áreas possíveis, como vendas e suporte.

Este trabalho apresenta uma discussão baseada num ambiente real, com características próximas da maioria das redes. Para possibilitar que uma equipe diminuta pudesse manter um nível satisfatório de qualidade de serviço numa rede de um Instituto de Pesquisas, com aproximadamente 400 (quatrocentas) estações, uma aplicação CBR foi implementada. Para sua base inicial, foram adquiridos casos que já vinham sendo documentados de forma textual e não indexada durante dois anos, acrescidos de casos simulados baseados na pesquisa à documentação de rede disponível. Para alcançar este objetivo, foi utilizado um *shell* para construção de aplicações baseadas em casos, o *CBR-Works* (WATSON, 1997). Este ambiente atende ao objetivo principal do trabalho, que é a obtenção da melhoria da produtividade na gerência de falhas em Redes, com redução do tempo médio de indisponibilidade. Visando contribuir para o estado-da-arte, foram idealizadas, implementadas e testadas duas ferramentas para evidenciar alguns aspectos fundamentais para a gerência de falhas que são ocultados pelos Sistemas Operacionais. Além disso, foi desenvolvida uma aplicação CBR genérica, cuja base pode ser utilizada em qualquer ambiente de Rede Local. Buscou-se manter, no entanto, a característica computacional *user-friendly* considerada chave para democratização do conhecimento, dentro do contexto da globalização e da inclusão digital. Os resultados mostram a viabilidade desta proposta. A Tese foi dividida da seguinte maneira: o restante deste capítulo detalha mais as motivações, antecedentes e propostas em discussão. O segundo capítulo ilustra os grandes desafios de um Gerente de Redes, as características das falhas e sua respectiva sintomatologia. O terceiro capítulo disserta sobre o diagnóstico de falhas em Redes de Computadores, focando especialmente no Raciocínio Baseado em Casos, eleito dentre os demais por permitir a absorção não só da experiência especialista dos técnicos de suporte, mas também do próprio usuário. O quarto capítulo descreve a metodologia de documentação INRECA (BERGMANN, GÖKER, 2003), que possui um conjunto de modelos de aplicações disponíveis. O quinto capítulo apresenta o desenvolvimento da aplicação para gerenciamento de falhas em Redes de Computadores com o uso do *CBR-Works*. No sexto capítulo as

funcionalidades da aplicação são ilustradas através de Estudos de Caso, respectivamente de falha na camada de Aplicação, de Rede e Física, para demonstrar o uso da metodologia de solução de falhas apoiada pelo CBR e pelas ferramentas de auxílio. O sétimo capítulo descreve os resultados e conclui a Tese.

1.1 Justificativa

Uma das áreas de pesquisa mais desafiadoras da Tecnologia da Informação e da Comunicação (TIC) é a área do Diagnóstico de Falhas em Redes de Computadores. A característica dinâmica das Redes dificulta a tarefa de se desenvolver modelos fiéis o suficiente ao perfil de um ambiente cuja possibilidade de mudanças é constante e a heterogeneidade de seus componentes é muito grande. Os sistemas CBR (Raciocínio Baseado em Casos – *Case-Based Reasoning*) são interessantes pelo aproveitamento do hábito dos gerentes de registrar ocorrências de falhas nos ambientes de redes, mas o crescimento do número de casos e a diferenciação das soluções possíveis decorrente de detalhes mínimos em cada caso tornam importante haver uma estratégia de otimização. Este capítulo se destina a fundamentar a proposta da construção de um CBR com base em um ambiente real e a avaliação de possíveis ferramentas de otimização em produção.

A ISO (*International Standardization Organization*) aponta como um dos maiores desafios na área de Gerenciamento de Redes de Comunicação de Dados o tratamento das falhas (LAZAR *et al*., 1992, CHUTANI, NUSSBAUMER, 1995, BARAS *et al*., 1997, GOPAL, 2000). Com o grande interesse pelas Redes de Computadores, incluindo-se as aplicações comerciais (*e-commerce*), e a evolução dos dispositivos móveis, a identificação e o isolamento das falhas são ao mesmo tempo mais complexos e mais importantes. As técnicas mais discutidas foram inicialmente desenvolvidas com foco em sistemas mecânicos, e mais recentemente tais conceitos vêm sendo estendidos para Redes de Computadores. Uma vez que a ISO prega a conectividade global de computadores e sistemas, através do seu modelo OSI (*Open Systems Interconnection*), cuja principal idéia é a do uso de arquiteturas abertas, urge a necessidade de se buscar soluções para o diagnóstico de falhas, integradas aos sistemas de gerenciamento. Desta forma, poder-se-ia garantir maior confiabilidade e Qualidade de Serviço (QoS) para as aplicações.

Apesar do gerenciamento de falhas ser considerado uma das mais importantes áreas de interesse pela ISO, apenas linhas gerais para formatos de notificações de

alarmes são padronizados, ficando a detecção e o diagnóstico das falhas por conta do responsável pela implementação. Essa abordagem torna o problema do gerenciamento particular, dependente dos dispositivos ou da configuração da rede, não havendo uma base para a integração e a montagem de um sistema de tolerância à falhas global. A existência de inconsistências em redes, como informações indisponíveis ou incompletas conduzem a condições cuja inferência para o diagnóstico é feita num cenário de muita incerteza.

Vários trabalhos têm apresentado propostas para solução destes problemas. Estas propostas são tipicamente associadas às áreas de Computação Tolerante à Falhas, Sistemas de Inteligência Artificial, Sistemas Especialistas e Teoria de Controle. Na área de Controle, metodologias de diagnóstico à base de modelos foram desenvolvidas nas décadas de 70 e 80 para sistemas lineares. Na década de 90, o interesse se voltou para sistemas não lineares. O problema da detecção da falha vem a ser, na maioria dos casos, um problema de avaliação do estado do sistema com base em informação "ruidosa". Esse modelo não se aplica bem ao gerenciamento de falhas em redes, principalmente porque as redes são modeladas como sistemas dinâmicos de eventos discretos onde a complexidade é uma questão chave. Neste contexto, o gerenciamento de falhas é usado para detectar, isolar e reparar problemas, distinguindo-se em quatro fases distintas:

- A *detecção* da falha, feita através do uso de alarmes gerados com base em estratégias de monitoramento;
- A *localização* da falha, possível através de um algoritmo que determine um conjunto possível de falhas;
- A *identificação* da falha, obtida através do teste de objetos hipoteticamente em falha e;
- A *correção* da falha, possível com a tomada de ações corretivas, em função do escopo definido nas fases anteriores.

A Computação Tolerante à Falhas tem contribuído para construção de propostas com algumas idéias, mas o ambiente distribuído dificulta sua implementação. Os sistemas especialistas têm sido muito usados nessa área, mas na sua maioria vem sendo construídos *ad-hoc*. A próxima seção discorrerá sobre alguns aspectos peculiares das falhas em Redes de Computadores e seus efeitos.

1.2 Falhas em Redes de Computadores e seus Efeitos

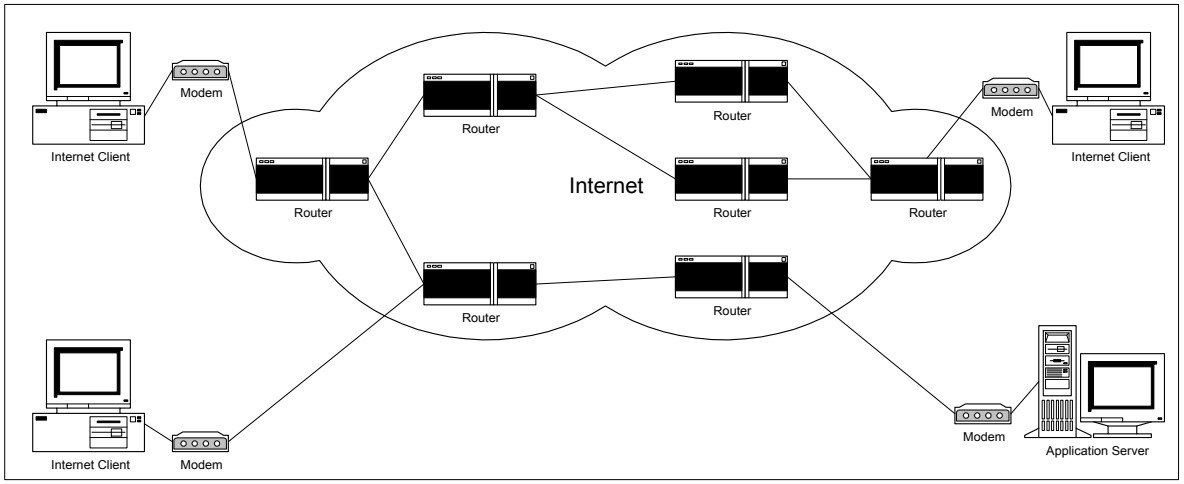


Figura 1 - Cenário Típico para ambientes Distribuídos

O cenário apresentado na figura 1 é um dos mais comuns arranjos topológicos de rede encontrados nos dias de hoje, para a maioria dos ambientes onde há conectividade. Máquinas clientes acessam remotamente um ou mais servidores, onde as aplicações e os dados estão disponíveis. O uso de uma arquitetura padrão de protocolos de comunicação, o TCP/IP, facilita a conectividade entre os equipamentos, independente de fabricante, sistema operacional ou capacidade de processamento. Para os enlaces entre os equipamentos, a complexidade pode ainda ser maior, considerando-se a possibilidade do uso de outras tecnologias, como o *Frame-Relay, X.25, xDSL, PPP, DocSis*. Podem-se ainda usar meios físicos diversos, como rádio microondas, fios de cobre, fibra ótica, satélites e outras. Suponha-se agora que um enlace falhe, como o que liga o servidor de aplicações à Internet. Essa única falha terá seus efeitos propagados. Uma aplicação poderá apresentar "*time-outs*" ("estouro" de tempo de espera para uma resposta) em requisições que passam por esse enlace, e novos avisos de erros serão reportados nos pedidos de outras conexões por esse caminho. As tabelas de roteamento serão atualizadas para eliminar o congestionamento naquele enlace, e os outros enlaces possivelmente ficarão sobrecarregados, passando a reportar degradação de performance. O resultado é que uma variedade de alarmes provenientes de várias entidades de gerenciamento (agentes) poderá "inundar" o centro de gerenciamento (gerentes) da rede. A tarefa difícil é a de determinar se o problema é temporário ou persistente, se múltiplos problemas ocorrerão simultaneamente ou se um problema teve efeitos propagados, e a exata localização do mesmo. O resultado final é que dois problemas

muito diferentes podem ter sintomas muito similares. O relacionamento entre as entidades gerenciadas ajuda a reduzir o escopo das possíveis fontes de falhas de acordo com os alarmes recebidos, e a estimar o grau de severidade da provável falha.

Os protocolos de comunicação usam *time-outs* e controles de erro em caso de falhas, para administrar tráfego. Essa estratégia implica diretamente na qualidade da rede, percebida no nível da aplicação. Alguns efeitos tipicamente observados dessas falhas são apresentados na tabela 1.

Tabela 1 - Falhas em Redes e seus Efeitos

Protocolo	Falha	Efeito observado
Transmissão de bits (camada física)	Colisões, demandando retransmissões.	"Lentidão" na rede
Transmission Control Protocol (Controle de fluxo – camada de Transporte)	congestionamento	Redução do throughput, "lentidão" na rede.
Internet Protocol (Fragmentação – Camada de Rede)	Perda de fragmentos, demandando descarte de todo o datagrama.	Redução do *throughput*, "lentidão" na rede.
Roteamento (Camada de Rede)	Falha em enlace (interrupção do caminho de transmissão)	Recebimento de mensagens de erro pouco úteis (*destination host unreachable*)

Pelos efeitos observados, é possível verificar que um mesmo sintoma (efeito observado) pode ser causado por falhas diversas, percebidas em camadas de protocolos diferentes.

O *Internet Protocol* (IP), usado mundialmente para interoperação entre Redes, pressupõe que suas unidades de dados, os datagramas, podem ser perdidos, duplicados, atrasados ou entregues fora de ordem. Há um mecanismo para detecção de erros ocorridos durante a transmissão, feito através do cálculo de *checksum* pelo receptor, mas, em caso de erro, o receptor apenas descarta o datagrama, sem nada informar ao emissor do mesmo. Esse mecanismo é decorrente da orientação básica do serviço prestado pela Internet, o *best-effort*. Essa orientação objetiva fazer o melhor que se puder para transmitir os dados, com o mínimo de tráfego possível. A falta de mensagens de erro explícitas para cada ocorrência dificulta a sua detecção.

A Arquitetura TCP/IP disponibiliza o protocolo ICMP (*Internet Control Message Protocol*) para relatar problemas que resultam em condições de erro. As mensagens do ICMP são criadas em resposta a um datagrama que sofreu alguma condição atípica, ou a uma mensagem que leve uma requisição do protocolo ICMP.

Para se evitar congestionamento na rede, seguindo os princípios do *best-effort*, mensagens ICMP que gerem erros não gerarão outras mensagens ICMP de erro. O ICMP, idealizado para informar erro e sua respectiva razão, não é adequado para diagnóstico de falhas, desassociado de outras ferramentas. É comum, por exemplo, o recebimento pelos usuários de mensagens do tipo "*destination host unreachable*" em seus navegadores (*Netscape*, *Internet Explorer*, *Mozzila Firefox*, etc), o que poderia sugerir, em uma primeira análise, que o servidor que responde pelo endereço requisitado está inacessível, por estar desligado ou com o seu enlace físico de acesso indisponível. Pela mensagem, o usuário poderia imediatamente imaginar problemas no seu computador, no Sistema Operacional, no modem, na linha telefônica, no provedor, etc. Na maioria das vezes, o uso da opção "atualizar" do navegador traz o recurso solicitado com sucesso, indicando que a verdadeira causa para a falha ocorrida na primeira tentativa foi, na verdade, um congestionamento temporário na rede que causou um estouro de *time-out*.

Os usuários, por sua vez, interagem com os sistemas conectados através de redes com aplicações cliente/servidor, e as mensagens de erro recebidas pela aplicação pouco ou nada ajudam no diagnóstico de falhas. A recusa de uma conexão a um domínio é tipicamente informada apenas com uma mensagem tipo "*login failed*", sem um mínimo de informação que permita uma análise das razões para a ocorrência da falha. Os administradores de rede, responsáveis pelo monitoramento, detecção e resolução de problemas, têm como primordial recurso a sua experiência, acumulada com a vivência entre os problemas e suas soluções *ad-hoc*. Uma opção para apoio a essa atividade é o uso do protocolo de aplicação SNMP (*Simple Network Management Protocol*), tipicamente implementado através de um aplicativo específico para gerência de redes. Esse gerenciamento, desenvolvido sob uma orientação Agente/Gerente, implementa funcionalidades semelhantes às dos sistemas de arquitetura cliente/servidor. O Gerente, operando como um Cliente, pode interrogar os dispositivos de rede onde estão sendo executados os Agentes SNMP, visando determinar o seu *status* e obter estatísticas sobre o seu funcionamento. O projeto SNMP é baseado em objetos, que são chamados coletivamente de MIB (*Management Information Base*). Esses objetos são usados para armazenar valores de variáveis, como contadores, por exemplo. A variável que contabiliza o número de datagramas IP recebidas por um dispositivo tem uma identificação do tipo:

iso.org.dod.internet.mgmt.mib.ip.ipInReceived

As Mensagens geradas através do ICMP formam dentro do SNMP um grupo com 26 variáveis padronizadas na RFC 1905 (disponíveis em http://www.rfc-editor.org/rfc/rfc1905.txt).

As variáveis MIB podem ser criadas e usadas livremente pelos sistemas de gerenciamento, o que provoca situações ambíguas: se por um lado, permitem a flexibilidade para adaptação dos sistemas de gerenciamento às peculiaridades das redes em particular, por outro lado dificultam a tarefa de diagnosticar uma falha em redes fim-a-fim. Esta dificuldade decorre do fato de que variáveis MIB usadas por um determinado sistema de gerenciamento podem ser completamente desconhecidas por Agentes SNMP de dispositivos de rede fora do ambiente local. Apesar de toda esta infra-estrutura de suporte ao diagnóstico e correção de falhas, a necessidade de tornar o mais simples quanto possível a visão do usuário final com relação à conectividade dificulta a sua adequada utilização. A próxima seção descreve algumas propostas para o tratamento de falhas em Redes.

1.3 Algumas Propostas em Discussão

Nesta seção, são citadas algumas propostas disponíveis na literatura. Deve-se observar que, na maioria delas, são encontradas características das várias abordagens buscando uma solução integrada que possibilite a implementação das vantagens de cada uma. Com isso, poderiam ser evitados os problemas que a pura e simples adoção de uma única estratégia traria. Algumas abordagens buscam usar recursos já padronizados e disponíveis nas redes, propondo apenas mecanismos para a utilização destes recursos.

O SNMP (*Simple Network Management Protocol*) é um protocolo de gerenciamento útil, porém pouco utilizado. Manipula variáveis MIB (*Management Information Base*) que armazenam informações importantes, como por exemplo erros de transmissão e *status* do enlace (*full* ou *half duplex*) que podem, por si só, permitir o diagnóstico de uma falha. Algumas propostas, como em DUARTE, DOS SANTOS, (2001), exploram o potencial do SNMP para o gerenciamento das falhas em Redes. Outro aspecto relevante é que as falhas não ocorrem de forma determinística, e nem sempre os sintomas percebidos são determinantes da falha em curso. Algumas propostas, como em THOTTAN, CHUANYI, (2000), buscam correlacionar a ocorrência de falhas com as estatísticas da rede, abstraindo-se das correlações de falhas.

Resultados mais consistentes que as propostas anteriores são observados nos trabalhos baseados em RBR (Raciocínio baseado em regras). Os mecanismos

suportados por RBR se baseiam em um conjunto de regras que definem como o sistema deve reagir de acordo com os sintomas decorrentes da entrada em um estado de falha. Esta abordagem, no entanto, não dá solução em casos para os quais não haja regras, nem possui um esquema para acompanhar a dinâmica típica dos ambientes de rede, com novos problemas e decorrentes soluções.

A Inteligência Artificial (IA) é observada em alguns trabalhos. Em LECKIE (1995), por exemplo, sintomas de falhas são extraídos da rede e submetidos a um sistema de diagnóstico especialista, para geração de um relatório de falha. Como é típico em sistemas deste tipo, a escalabilidade não é uma de suas virtudes. Em LECKIE *et al*. (1997), essa função é distribuída para contornar sua inabilidade escalar e, principalmente, o problema da propagação de alarmes causados por falhas, que eventualmente causam sintomas diferentes dos causados originalmente pela falha inicial. Os Sistemas Especialistas tipicamente empregam o raciocínio dedutivo, gerando soluções apenas para o conhecimento disponível, mas há sugestões empregando o Raciocínio Abdutivo, onde os resultados são extrapolados até todas as possíveis falhas, buscando-se, em uma via inversa em direção aos sintomas, em busca do resultado com a maior confiança.

O CBR (Raciocínio Baseado em Casos) busca exatamente suprir a principal deficiência do RBR, que é a de não permitir uma contínua absorção do conhecimento especialista disponível, na forma de casos, e não de regras. Além disso, possibilita que um caso inserido no sistema seja "adaptado" para a busca de um diagnóstico de falha apropriado. Isto é interessante, porque nem sempre todos os sintomas são descritos, apesar de estarem presentes na falha em curso. Em LEWIS (1993) algumas métricas são estabelecidas para descrição dos sintomas, que, com maior ou menor relevância, vão definir quais são os casos disponíveis mais similares. O sistema permite ainda que valores diferentes de parâmetros descritores de características da rede sejam usados em funções, *adaptando* um caso disponível ao submetido.

As Redes Neurais também estão presentes nesta disciplina. A idéia é que a Rede Neural possa tratar as informações da rede, inferindo possibilidades de falhas, que aí sim são submetidas a um sistema especialista. Esta abordagem flexibiliza o cenário encontrado no RBR, por permitir o tratamento de novos casos, mas é sujeita à indicação de falso-positivos em caso de correlação de alarmes e propagação de falhas.

Em virtude da característica lingüística da descrição de sintomas de falhas, a Lógica *Fuzzy* é base para algumas propostas. Em CHEN, HUANG (1996), um Sistema

Fuzzy é usado na interface com o usuário para capturar a melhor descrição dos sintomas, e com um histórico de falhas, para incorporação a uma base de conhecimento. Assim, é possível indicar graus de pertinência para dispositivos em falha e para ações de recuperação, permitindo a opção de solução de forma priorizada.

A probabilidade também é base para vários trabalhos, como em BRODIE *et al*., onde valores de parâmetros de rede são obtidos através de *probes*, e os resultados coletados são submetidos a uma Rede Bayesiana que infere um diagnóstico.

O RBM (Raciocínio Baseado em Modelos) tem sido usado para propostas (HUARD, 1993, ROUVELLOU, HART, 1995, DALMI *et al*., 1998, CHAO *et al*. , 1999, CHAO *et al*., 2001) que se fundamentam em Máquinas de Estados Finitos (MEF), porém, nem sempre a ocorrência de uma falha (como lentidão causada por tráfego excessivo) causará uma transição incorreta. As MEF podem ser expressas através de Redes de Petri, e estas, se temporizadas, podem capturar falhas deste tipo. No entanto, os modelos não escalam e não são dinâmicos como os ambientes de rede. Em ROUVELLOU, HART (1995), as MEF tem agregadas a si uma componente probabilística, para busca da solução para os alarmes correlacionados.
Estas propostas serão apresentadas mais amiúde no capítulo 3.

A adoção de um mecanismo único e padronizado é bastante desejável, para possibilitar uma ampla conectividade entre as redes. Para isso, é fundamental que as propostas busquem, tanto quanto possível, usar as ferramentas de monitoração de falhas já disponíveis, como o ICMP. Devem também ser mais genéricas e abrangentes quanto possível, oferecendo a máxima rapidez e precisão nos diagnósticos. Tal requisito se justifica pela tendência cada vez maior de interligação entre dispositivos heterogêneos numa única infra-estrutura, como na Internet. Como já se observa em algumas propostas, o uso de agentes distribuídos com protocolos de comunicação padronizados entre outros gerentes pode resolver o problema da escalabilidade e das especificidades de um ou outro ambiente. A combinação de estratégias, como as abordagens por modelos e por regras, também é uma alternativa. Para modelagem, as Máquinas de Estados Finitos (MEFs) são a alternativa mais imediata, uma vez que as mesmas são extensivamente usadas no projeto de protocolos.

1.4 Motivações e Principais Contribuições do Trabalho

Grandes redes locais, por mais bem implementadas que tenham sido, dependem de uma competente equipe de suporte para administrá-las. Tipicamente, com o

crescimento da rede, os problemas de conectividade se propagam e causam sintomas que afetam vários equipamentos. Em paralelo a isso, tais sintomas normalmente não são muito expressivos, dificultando a tarefa de diagnosticar a falha e causando grande dependência do conhecimento especialista da equipe de suporte.

O Instituto de Pesquisas da Marinha possui uma rede local com aproximadamente 400 nós. A equipe de suporte (*staff*), em função da grande carência de pessoal, confunde-se com a equipe de desenvolvimento, em sistema de rodízio. Para permitir um mínimo de interrupção do trabalho da equipe de desenvolvimento, foi desenvolvido um sistema de relatos de falhas e suas soluções, chamado de SOS. Neste sistema, cada técnico, quando envolvido em tarefas de suporte, descreve todas as características dos problemas enfrentados no seu turno e suas eventuais soluções, de forma textual. No decorrer do desenvolvimento deste trabalho, o sistema já contava com aproximadamente 60 registros, e, apesar do conhecimento estar descrito nos casos, é perceptível que a solução para o registro e acesso textual de conhecimento especialista não é escalável, pois não há tratamento automatizado para as similaridades entre os casos atuais e os passados. Essa restrição causa a necessidade da busca seqüencial entre os casos, permitindo apenas uma segmentação de grupos de casos de acordo com o assunto da falha, que pode ser um problema num determinado sistema ou na rede em si. Muitas vezes, observa-se que o que o analista de suporte inicialmente identificou como um problema num sistema, na verdade trata-se de um problema de conectividade.

1.4.1 Motivações

As opções deste trabalho foram orientadas por várias motivações. Dentre elas, pode-se citar:

- O desafio diário do autor para administrar um ambiente híbrido, cujas falhas ocorrem independentemente da disposição da equipe de suporte para mantê-la funcional;
- A semelhança entre os ambientes de rede, que motivam esse trabalho de pesquisa para a minimização do tempo de indisponibilidade das redes, por maiores que sejam;
- O aprofundamento em uma disciplina de grande utilidade para o caso em foco, o CBR, devido a sua capacidade de incorporar conhecimento especialista de forma dinâmica e natural; e

- O desafio de buscar soluções para aprimorar o modelo típico de operação do setor de *help-desk* das redes, onde o conjunto de sintomas é a única fonte de busca inicial para o diagnóstico. Esta é uma das principais causas para o longo tempo de indisponibilidade das estações de trabalho das redes.

1.4.2 Principais Contribuições do Trabalho

No decorrer do trabalho, foram alcançados alguns resultados relevantes, dentre os quais podem ser citados os seguintes:

- O desenvolvimento de uma Aplicação CBR com base em dados genéricos, típicos de qualquer rede atual, enriquecida com casos reais observados no *test-bed* disponível, igualmente de caráter geral e aplicáveis à qualquer ambiente de Rede Local;
- O desenvolvimento de uma Aplicação para captura e análise de tráfego ICMP, tipicamente não usado para detecção automática, por exemplo, de erros na configuração do endereço da estação de trabalho e a existência de falhas de segurança, entre outras;
- O desenvolvimento de uma Aplicação para captura remota do ambiente de conectividade local configurado na estação de trabalho, independente do Sistema Operacional em uso, reduzindo o tempo de diagnóstico em vários casos de falhas e melhorando a descrição dos sintomas dos casos;
- A documentação da Aplicação CBR desenvolvida através de um padrão usado por vários grupos de pesquisa (INRECA), possibilitando a agregação de conhecimento ao estado-da-arte e continuidade da pesquisa; e
- Testes em ambiente real de produção, com a consolidação dos resultados em termos de redução da indisponibilidade média, visando comprovar a viabilidade da proposta deste trabalho.

Este capítulo se propôs a apresentar a proposta para Gerenciamento de Redes de Computadores com o uso do Raciocínio Baseado em Casos e Ferramentas auxiliares. O próximo capítulo descreve os principais desafios enfrentados pelos Gerentes de Redes, em virtude das heterogeneidades de equipamentos e sistemas, correlações de alarmes, propagação de falhas e, principalmente, urgência na solução de problemas.

2 Desafios no Gerenciamento de Redes

O Gerenciamento de Redes de Computadores é uma tarefa desafiadora. Os sinais e sintomas nem sempre indicam claramente um diagnóstico preciso, e são típicas as situações onde o conhecimento especialista de um profissional é determinante na solução de problemas, principalmente por já ter passado pelos mesmos. Via de regra, tal comportamento, apesar de atender imediatamente aos anseios das corporações usuárias das redes em mantê-las operacionais, causa grande dependência pessoal de um determinado profissional, o que não é desejável. O ideal é que este conhecimento especialista, principalmente obtido através de casos semelhantes enfrentados no passado ficasse disponível para outros profissionais, possibilitando a sua utilização por qualquer um sempre que necessário. Outro aspecto é que muitos problemas apontados por usuários são de resolução trivial, mas que, pelo fato do usuário não possuir conhecimento especialista algum, o mesmo ocupa precioso tempo de profissionais de *help-desk*, que poderiam estar exercendo atividades mais nobres como monitorar a rede e fazer ajustes para melhoria de performance. Este capítulo apresenta aspectos práticos do Gerenciamento de Redes, que servirão como fundamento para a construção do modelo a ser adotado para o CBR, bem como para a inserção dos primeiros casos onde o diagnóstico, com base nos sintomas e sinais, é trivial.

2.1 Fundamentos do Gerenciamento de Redes

A missão de um Gerente de Redes, através de sua equipe, é monitorar e controlar elementos de rede com o objetivo de assegurar um nível satisfatório de Qualidade de Serviço. Este nível normalmente pode ser definido teoricamente, de acordo com o número de estações e as tecnologias utilizadas, mas a sensibilidade do usuário é o principal alarme. Não que ele conheça tais limites teóricos, mas principalmente pela sensação de mudança comportamental das taxas de comunicação entre a sua estação e as demais da rede. O Gerente possui a sua disposição toda uma série de ferramentas de monitoramento, gratuitas ou não, visando obter informações privilegiadas para a elaboração de diagnósticos. Estas ferramentas operam, na sua maioria, segundo o modelo cliente-servidor, o que na área de gerência é chamado de agente-gerente. Nessa abordagem, o gerente é um software que possibilita a captura de valores de parâmetros junto aos agentes, que por sua vez são chamados de elementos gerenciados quando possuem o software de armazenamento dos parâmetros citados. A conversação entre

estes dois elementos é regida por um conjunto de regras e formatos chamado de Protocolo de Gerência. Os protocolos padrão de gerência tipicamente usados nas redes são o SNMP (*Simple Network Management Protocol*) CASE *et al.* (2006) e o ICMP (*Internet Control Message Protocol*) POSTEL (2006), e, por isso, este trabalho focará neles. Várias são as informações passíveis de obtenção através do relacionamento agente-gerente SNMP, como taxas de erros, status de operação de interfaces e equipamentos, taxas de utilização de interfaces, protocolos em operação e inúmeras outras. Estas informações são preciosas na atividade de Gerenciamento, porém, apenas são úteis para quem sabe interpretá-las. A tentativa de conexão despropositada também é percebida trivialmente pelo ICMP, mas nem por isso é notificada ao usuário. O conhecimento especialista permite, por exemplo, a determinação de uma taxa máxima (*threshold*) de erro para um determinado enlace, de acordo com a sua taxa normal de operação.

Para a modelagem da aplicação CBR deste trabalho, será adotada, com algumas adaptações baseadas na experiência pessoal do autor deste trabalho, a metodologia de detecção, diagnóstico e resolução de problemas descrita em LOPES *et al.* (2003). Nesta abordagem, um problema (ou, para os objetivos deste trabalho, um caso), é definido através dos seguintes elementos:

- Descrição – circunstâncias típicas nas quais o problema ocorre; causas mais comuns e subconjuntos específicos do problema;
- Sintomas – características específicas perceptíveis durante a ocorrência do problema;
- Sinais – sutilmente diferentes dos sintomas, por representar características mais internas da rede em estado alterado, em conseqüência da ocorrência de um problema. Estes elementos paramétricos não são tipicamente perceptíveis para os usuários comuns, uma vez que tais elementos são obtidos através de ferramentas de auxílio ao gerenciamento. Por se tratar de manifestações adicionais, devem ser buscados com o objetivo de enriquecer o caso e possibilitar uma maior precisão no diagnóstico;
- Testes confirmatórios – descrevem passos a serem realizados com o objetivo de confirmar ou negar um diagnóstico específico. Não são necessários caso tenham sido obtidos sinais diferenciais para a confirmação de um determinado tipo de falha específica; e

- Sugestões de Tratamento – possível solução que, sem causar novos problemas na rede, resolva especificamente o problema descrito.

Outro detalhe importante é que serão focados exclusivamente problemas relativos às redes Ethernet e suas variações, uma vez que são mais presentes do que as redes *token-ring*, *AppleTalk* e outras.

2.2 Metodologia para Detecção, Diagnóstico e Resolução de Problemas

A primeira indicação de que algo anormal está ocorrendo na rede é a percepção de algum sintoma. Este evento, chamado de detecção, pode ser notificado por um usuário, um sistema, através de alarmes ou a simples exposição gráfica da evolução de um determinado parâmetro, ou ainda pela verificação pelo pessoal de suporte da rede de que algo errado está acontecendo. Nessa última hipótese, *leds* indicativos de operacionalidade costumam ser bastante úteis. A metodologia adotada pode ser descrita graficamente segundo o fluxograma ilustrado na figura 2.

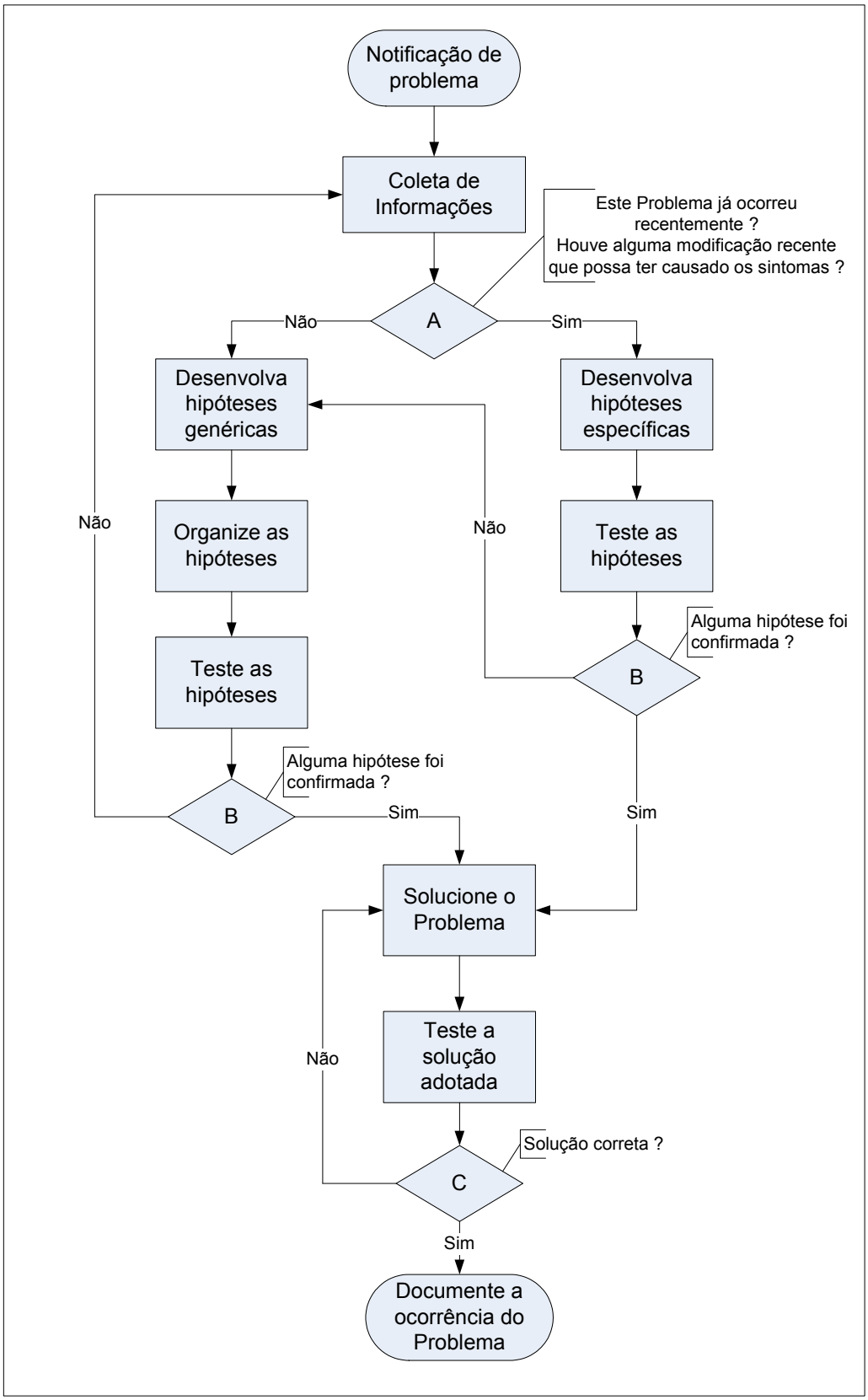


Figura 2 - Fluxograma da metodologia para resolução de problemas em redes

Durante a fase de detecção, a disponibilidade de informações mais refinadas permitirá a obtenção de diagnósticos mais precisos. Por conta disso, é proposta nesse trabalho a utilização de um *plug-in* a ser instalado junto ao sistema operacional do computador,

para obtenção de dados importantes para o diagnóstico, porém que não são normalmente de conhecimento dos usuários, visando minimizar a ocorrência de chamados ao *help-desk* com o uso da ferramenta CBR. Na forma convencional, cabe aos técnicos obter respostas para algumas questões importantes como;

- Qual é o universo de estações, equipamentos de conectividade e segmentos de rede atingidos pelo problema ?
- Quais são as características temporais do problema (freqüência, manifestação, início da sua percepção) ?
- Há algum evento que esteja relacionado com o problema, ou seja, ele apenas se manifesta quando alguma estação é ligada, ou algum programa é executado ?
- Há algum alarme referente à ocorrência ? Qual é a interpretação da mensagem de erro gerada ?
- Os parâmetros da rede, obtidos através de ferramentas, estão dentro da normalidade ? Há conectividade entre a estação com problema e os dispositivos de conectividade principais como o *switch* local, o *switch* do *backplane*, seu *gateway*, outros importantes ?
- Foi feita alguma modificação na rede que possa ter influído na geração do problema manifestado ?
- Este problema tem semelhança com algum outro recente ?

Nessa última questão é que se propõe o uso de uma aplicação CBR. Grandes redes normalmente experimentam os mesmos problemas, cujas soluções ficam concentradas apenas numa pequena equipe que memoriza as mesmas. Outra questão é que são importantes para o desenvolvimento das hipóteses de causas para os problemas: o conhecimento dos protocolos de comunicação, a topologia local e os métodos de acesso presentes nas redes. A indisponibilidade destes profissionais pode retardar a volta da rede ao seu estado normal e com isso reduzir suas funcionalidades. Um CBR contendo todos os casos ocorridos e suas soluções é uma poderosa ferramenta de documentação que indicará caminhos de solução (hipóteses) ou até mesmo a solução mais indicada, no caso de similaridade de sintomas e sinais. A funcionalidade da adaptação presente no CBR poderá ser muito útil, na medida em que vários problemas são parametrizados em função de números de estações em segmentos e outras variáveis, que podem estar inseridas no sistema e adaptar um caso para a situação em estudo, minimizando o tempo para a solução.

A experiência recomenda segmentar as categorias de problemas segundo o modelo OSI, uma vez que a grande maioria dos problemas em Redes Locais ocorre em decorrência de irregularidades na camada física/enlace, e a seguir na camada de rede (CHAPPELL, FARKAS, 2003). Tipicamente, as aplicações de diagnóstico se concentram nas camadas de enlace e de rede, em função dos equipamentos de conectividade disponíveis nas redes (*switches* e roteadores). Há estatísticas disponíveis que afirmam que 90% dos problemas de uma rede são causados por falhas no cabeamento (HAUGDAHL, 2000). A figura 3 ilustra o modelo OSI e sua relação com as redes, ou seja, suas funcionalidades e dispositivos típicos de conectividade. As camadas 1 (física), 2 (enlace) e 3 (rede) estão destacadas por representar a quase totalidade de ocorrência de problemas, com maior concentração na camada física.

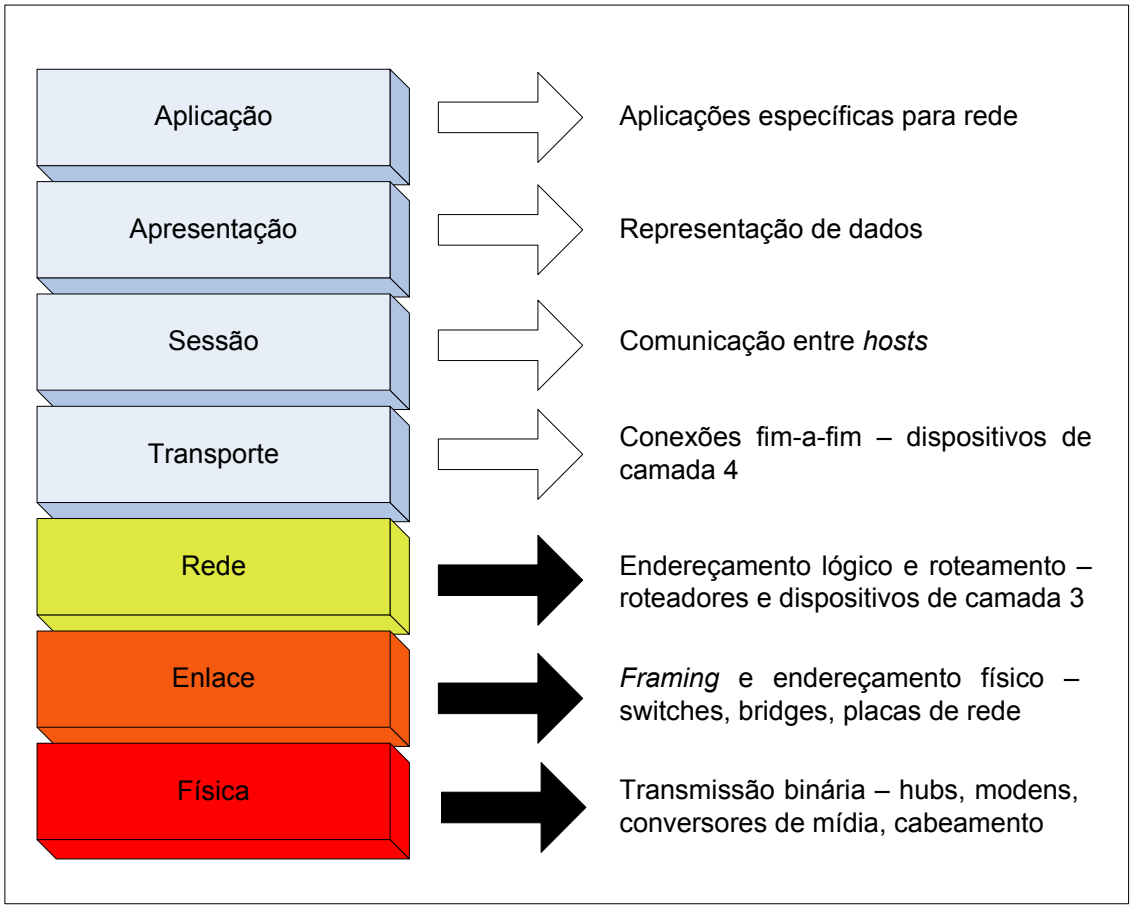


Figura 3 - Modelo OSI e sua relação com as redes

De posse dos sintomas e sinais, algumas hipóteses já podem ser formuladas. Estas devem ser testadas, iniciando-se pela camada física. Um sintoma de falta de conectividade, por exemplo, deve inicialmente ser analisado através da verificação da correção das configurações de rede (neste trabalho, feito através de uma ferramenta

própria). Uma vez descartada a hipótese de configuração errada, com o uso das informações coletadas pela própria ferramenta, a continuidade física e lógica pode ser testada com uma aplicação presente em todos os Sistemas Operacionais baseados em TCP/IP, o *ping*. Esta aplicação, tipicamente disponível na forma de comando do *Shell*, nada mais faz do que gerar uma mensagem ICMP (*Internet Control Message Protocol*) (POSTEL, 2006) do tipo *echo* para um determinado endereço de destino, que a responderá com *echo reply*. Se a resposta for obtida, problemas de camadas inferiores podem ser descartados, indicando possíveis problemas da camada 4 em diante.

Uma solução testada, com sucesso ou não, deve ser documentada para enriquecimento da Base de Conhecimento. Cabe ressaltar a importância de monitorar a rede após a implementação da solução, para observação se a mesma não introduziu outros problemas, como aumento no tráfego na rede ou pacotes espúrios na rede.

Na próxima seção, serão explanados alguns dos problemas típicos de redes , seus sintomas e sinais, e possíveis soluções. Estes problemas serão introduzidos na aplicação CBR como base de conhecimento pré-existente, assim como alguns problemas provocados intencionalmente para enriquecer essa base. De acordo com a experiência do autor, como a maior incidência de problemas ocorre na camada física, seguindo-se pela de enlace e a de rede, essa será a ordem de apresentação dos mesmos. A camada de aplicação, por ser mais específica e ser responsável por uma menor parcela de falhas, será abordada apenas em ocorrências pontuais, como é o caso das contaminações virais.

2.3 Problemas Típicos em Redes de Comunicação de Dados

Apesar das inúmeras possibilidades de combinação de equipamentos de conectividade para a criação de uma rede, os problemas podem ser isolados o suficiente para que possam ser generalizados. Assim, independentemente da rede que se esteja gerenciando, os mesmos casos de falhas podem ser usados para diagnóstico e solução de problemas. Para viabilizar uma dos objetivos contributivos deste trabalho, que é a definição de um conjunto de falhas típicas em redes, com base em seus sintomas em sinais, foram catalogados os principais problemas, através de documentação técnica e a experiência especialista do autor. Estes problemas estão detalhadamente descritos no apêndice 1, e são sumarizados na tabela 2.

Tabela 2 - Sumário das Principais Falhas em Redes

Camada OSI	Falha	Principais Sintomas
Aplicação	Contaminação Viral	(1)
Rede	Default GW incorreto	(2), (4)
	End. IP incorreto	(2), (3)
	Máscara de rede incorreta	(3)
	End. DNS incorreto	(2)
Enlace	Interface administrativamente desabilitada	(2)
	Tempestade de *broadcasts*	(1), (2), (3)
	TTL no comutador inadequado	(1), (3)
	TTL do *Cache* ARP inadequado	(1), (3)
Física	Cabo seccionado	(1), (2), (3)
	Conectores irregulares	(1), (2), (3)
	Incompatibilidade de Modo/Tx. operação	(1), (2)
	Equipamento defeituoso	(1), (2), (3)
	Interface defeituosa	(1), (2), (3)
	Interferências do ambiente	(1), (3)
	Congestionamento no barramento	(1)
	Cabo incorreto	(1), (2)
	Descumprimento das regras Ethernet	(1), (2)

Legenda: (1) Rede Lenta; (2) Falta de Conectividade; (3) Conectividade Intermitente; (4) Falta de conectividade apenas para fora da Rede Local (ex.: Internet).

A simples interpretação desta tabela nos permite concluir o seguinte: Como se pode observar na descrição de problemas típicos de redes enumerados, o alto nível de abstração oferecido para os usuários torna a tarefa de gerenciar desafiadora, uma vez que a grande maioria dos problemas será relatada através de informações subjetivas como: “a rede está lenta”, ou então “a rede está fora”; apesar de terem sido apresentados apenas os problemas mais comuns, a ocorrência de problemas é crescente da camada de aplicação para a camada física, ou seja, a grande maioria dos problemas de rede é decorrente de problemas na infra-estrutura; e que a descrição, por parte de um usuário, de que “a rede está lenta”, sob sua perspectiva, não nos permite nenhuma decisão imediata de solução, já que a maioria dos problemas possuirá tal sintoma. Em função disso, parece pertinente a idéia de disponibilizar ferramentas fáceis de usar que

permitam ao próprio usuário o enriquecimento da descrição do caso. O aplicativo de captura e análise de pacotes ICMP, por exemplo, possibilita a um usuário a identificação de uma possível contaminação viral. Já o programa de descrição do ambiente de rede possibilita a descoberta de parâmetros da rede (como endereços) que facilitam, por exemplo, a identificação de todas as falhas da camada de rede descritas na tabela 2.

O objetivo deste capítulo foi apresentar os principais desafios para o Gerenciamento das Redes, descrevendo alguns problemas típicos de forma genérica. Essa descrição é útil para inserção dos casos iniciais da aplicação CBR desenvolvida, possibilitando assim a sua modelagem, bem como a orientação para inserção de novos casos, de acordo com a experiência diária.

3 Diagnóstico de Falhas em Redes de Computadores

Diagnosticar falhas em Redes de Computadores não é trivial, por conta da similaridade entre os sintomas das diversas falhas. Este capítulo descreve algumas abordagens encontradas na literatura para viabilizar esta tarefa.

3.1 Principais Abordagens no Diagnóstico de Redes

Em virtude das dificuldades relacionadas com a administração de falhas em redes, várias propostas podem ser encontradas. A seguir, algumas estratégias para abordar o problema serão explanadas.

3.1.1 Raciocínio Baseado em Regras (RBR)

É o fundamento para a maioria dos trabalhos (LAZAR *et al.*, 1992, MAEDA, 1992, LECKIE, 1995, KUMAR, VENKATARAM, 1995, LECKIE *et al.*, 1997, DALMI *et al.*, 1998, GOPAL, 2000). O conhecimento é categorizado como geral ou específico. O conhecimento geral é expresso na forma de regras, e o específico atende a uma determinada situação particular; é constituído de fatos expressos através de assertivas. Um engenho de inferência possui um importante papel no raciocínio, por combinar fatos com regras para inferir nova informação a ser agregada. Essa técnica tem a vantagem de ser representada de forma similar ao pensamento humano (regras tipo: *se, então*), facilitando a captura do conhecimento. É modular, possibilitando a revisão, verificação e correção de regras sem interferir em outras já formuladas. É usada em casos de conhecimento impreciso ou incompleto. Como limitação principal, pode-se apontar que apenas na perfeita concordância com os antecedentes das regras (se) é possível direcionar o sistema para inferir nova informação. Há relacionamentos lógicos implícitos entre regras de difícil percepção. Também não usa experiências passadas no processo dedutivo. Com o aumento do número de regras, aumenta o tempo necessário para a decisão, tornando o método inviável para aplicações de tempo real. Estes sistemas consistem numa memória de trabalho, uma base de regras e um procedimento de controle. Para uma aplicação de redes, a base de dados contém a representação das características da rede, incluindo informações de topologia e de estados. A base de regras representa o conhecimento sobre que operações devem ser executadas quando o sistema entra num estado indesejável. Neste caso, o procedimento de controle seleciona as regras aplicáveis para a situação. Das regras selecionadas, uma estratégia de controle

é usada para definir que regra será efetivamente executada. Essa regra pode executar testes, fazer buscas em bases de dados, executar diretivas de gerência de configuração da rede, ou até mesmo invocar outro sistema especialista. Com estes resultados, o sistema atualiza a sua memória de trabalho e seus elementos. Para construir um sistema segundo esse paradigma, deve-se inicialmente definir uma linguagem de descrição que represente o domínio do problema; extrair conhecimento de técnicos e/ou de documentos que descrevam problemas e suas soluções; e representar essa experiência no formato RBR escolhido. Esse modelo falhará quando submetido a novos problemas, além de não ter a capacidade de adaptar o conhecimento existente às novas situações ou aprender com as experiências durante sua execução. Essas características tornam o RBR uma estratégia mais adequada para ambientes estáticos e de comportamento bem definido, o que não é o caso das grandes redes de comunicação.

É muito comum a associação de técnicas de Inteligência Artificial (IA) em 3 níveis: monitoramento da performance, diagnóstico de falhas, controle e configuração de redes. Estes níveis são naturalmente hierárquicos, e em todos eles algum grau de análise especialista é envolvido, tornando-os candidatos naturais para o uso de IA. Para que tais técnicas sejam mais facilmente adotadas pelos usuários, entende-se que se introduzam os mecanismos de IA inicialmente pelos níveis mais baixos. Verificando um exemplo de uma aplicação projetada para automatizar as tarefas de monitoramento de performance e diagnóstico de falhas em equipamentos de transmissão em redes telefônicas, observamos que o grande volume de dados torna impossível a análise de todos os aspectos da rede. Para simplificar e viabilizar essa tarefa, foi proposta em LECKIE (1995) a arquitetura mostrada na figura 4:

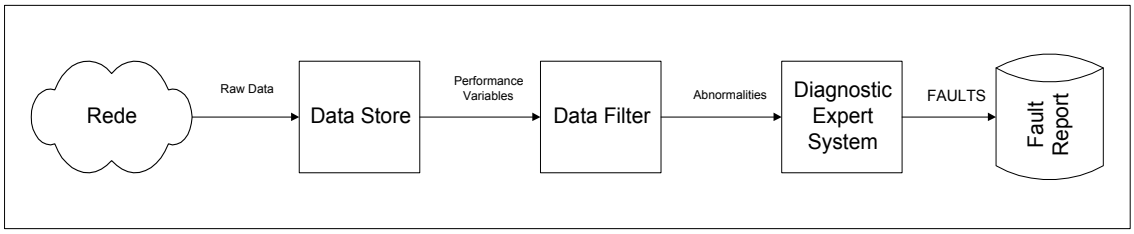


Figura 4 - Arquitetura para filtragem de falhas

O primeiro elemento dessa arquitetura é o filtro, que seleciona apenas os dados que sejam passíveis de indicar falha. Este filtro é definido pelos especialistas. Uma vez que o filtro contenha o conhecimento necessário, o sistema especialista, baseado em regras, executa o diagnóstico de acordo com a saída do filtro. Neste elemento está contida

também a correlação entre anormalidades dos vários elementos da rede. As saídas do sistema são usadas então para as decisões sobre correções de falhas.

3.1.2 Raciocínio Probabilístico

Na lógica clássica, as proposições podem ser falsas ou verdadeiras, o que não representa o comportamento do mundo real. A abordagem probabilística agrega uma probabilidade a cada proposição, para exprimir um determinado nível de incerteza. As redes *Bayesianas* são bastante usadas nas propostas disponíveis na literatura, conforme será visto na próxima seção. Uma rede *Bayesiana* é um grafo acíclico, direcionado, no qual *nós* representam um conjunto de variáveis e arcos conectam pares de *nós*, significando uma relação de causa ou de influência. Uma seta do *nó* X para o *nó* Y, significa que X tem influência direta em Y. Cada *nó* tem uma tabela de probabilidades condicionais, que quantifica os efeitos que os *nós* predecessores (pais) têm sobre um determinado *nó*. O sistema computa a probabilidade na forma de eventos P(*query*/evento), de forma que se possa decidir que ação tomar.

Entre as várias estratégias de possível aplicação, as Redes *Bayesianas* possuem grande aceitação. Sua implementação é interessante onde se precisa modelar uma situação na qual há uma relação de casualidade, mas o entendimento do que está acontecendo é incompleto, de forma que é necessário descrever tal fenômeno de forma probabilística. As Redes *Bayesianas* possibilitam calcular probabilidades condicionais de *nós* (estados, evidências, alarmes, ocorrências) uma vez que estejam disponíveis os valores observados em outros *nós*. Num grafo de casualidade como o exemplo da figura 5, é possível, ao recebimento de uma informação que o acesso falhou (AF) e a ocorrência de um alarme do link (LA), calcular as probabilidades condicionais da falha no link (P (LF | AF,LA)), congestionamento (P (NC | AF,LA)) e falha no servidor (P (SF | AF,LA)).

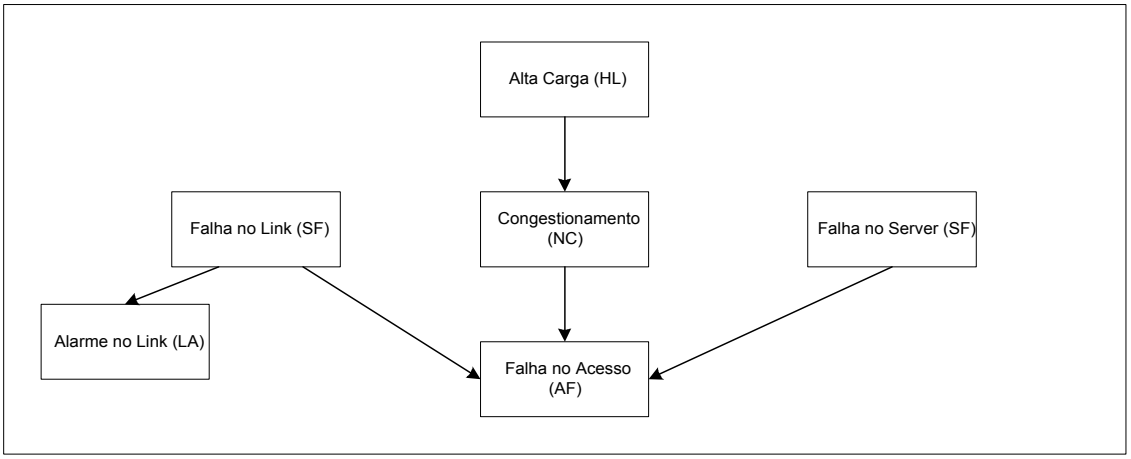


Figura 5 - Grafo Causal (extrato) de uma Rede Ethernet

Essa probabilidade condicional de um nó, dadas as evidências disponíveis, é chamada de *belief function* do nó. O diagnóstico com Redes Bayesianas, então, é o processo de capturar evidências e de calcular as *Belief functions*. Para redes representadas por grafos com múltiplas conexões, há estratégias como o *clustering*, onde há a formação de redes de *clusters* que são conectadas. Se os clusters forem grandes, entretanto, a computação das *belief functions* será complexa e demorada.

3.1.3 Raciocínio Baseado em Modelos (RBM)

Nesta técnica, o conhecimento é representado como um conjunto de modelos. A resolução dos problemas é feita através de profundo entendimento do domínio. No RBR, as regras são geradas de dados empíricos, e o sistema é uma caixa-preta. O CBR, novo conhecimento gera novas entradas, mas o sistema continua ignorado. Em contraste com estas abordagens, o RBM requer a construção de modelos para representar o sistema. Cada modelo é derivado da percepção comportamental do sistema em várias situações. Essa abordagem possibilita a construção de modelos gerais para o sistema, capazes de diagnosticar falhas que nunca ocorreram antes. Possui, entretanto, alguns dilemas, como por exemplo: se o sistema não for precisamente entendido, o modelo terá problemas de validação. Há algumas questões básicas em aberto nessa área de pesquisa: Como uma rede de comunicação de dados deve ser modelada para o gerenciamento de falhas ? Qual seria a estrutura fundamental do problema do gerenciamento de falhas sobre os vários modelos existentes? Que procedimentos devem ser assumidos para detecção e identificação de falhas ? São documentadas duas formas para resolução destas questões. Numa delas, a rede é abordada numa perspectiva global, o que conduz a modelos e algoritmos probabilísticos (MAEDA, 1992, CHOUDHURY, 1999,

BRODIE *et al.,* 2002a, BRODIE *et al.,* 2002b). Na outra, as entidades da rede ou as falhas são modeladas detalhadamente através de Máquinas de Estados Finitos (MEFs) (LAZAR *et al.,* 1992, CHAO *et al.,* 2001a, CHAO *et al.,* 2001b). Algumas propostas introduzem as probabilidades na própria MEF (ROUVELLOU, HART, 1995).

As MEFs são extensivamente usadas no projeto de protocolos. Estas ferramentas representam as alterações de estados nos protocolos de comunicação através de uma mudança de estado em uma MEF, e qualquer saída associada com uma alteração de estado no protocolo corresponde a uma entrada na MEF. Como normalmente apenas estas saídas são observáveis, são tipicamente chamadas de eventos. Por outro lado, uma vez que elas correspondem a entradas para a MEF, elas são chamadas de alfabeto de entrada. Sob o ponto de vista do observador, a detecção e o diagnóstico da falha requerem a verificação de se a rede está se comportando de acordo com a sua MEF, e, caso negativo, como ela se desvia do comportamento esperado. A tarefa, então, inclui aspectos como a construção de um alfabeto mínimo para detecção de falhas numa MEF, e como identificar a alteração na MEF quando a falha ocorreu.

Cabe ressaltar que a tarefa de modelar falhas com o uso das MEFs traz consigo toda a complexidade da explosão de estados encontrada em situações reais.

A conversão de MEFs para Redes de Petri é trivial, e tipicamente é utilizada pelos projetistas de protocolos para as tarefas relacionadas com a verificação, a validação e testes com os protocolos especificados. Essas redes são especialmente interessantes para a modelagem de sistemas distribuídos, com inúmeros eventos concorrentes, como é o caso dos alarmes de erros. A fundamentação matemática das Redes de Petri permite a construção de algoritmos para o gerenciamento de falhas, com potencial bastante adequado para a modelagem dos eventos em uma rede de comunicação de dados.

Um exemplo de proposta baseada em RBM pode ser visto em CHAO *et al.* (1995). Seu modelo é orientado a domínios hierárquicos, combinado com delegação de gerenciamento, o que lhe confere escalabilidade. Para modelar as falhas, define-se o seguinte vetor:

$$<\{<a_i, p_i>\}\ name\_F_j, \Delta t_k, layer_{\ l}, \{<s_m, w_m>\}>$$

Neste vetor, a_i é um alarme, p_i é uma probabilidade condicional, $name\_F_j$ é o rótulo da falha, Δt_k é o intervalo de tempo de ocorrência da falha, $layer_{\ l}$ é a camada de rede onde

a falha ocorre, s_m é o segmento físico da mesma, e w_m é um "peso" da ocorrência de F_j neste segmento. Por exemplo: {<a_1, 0,9>, <a_2, 0,2>}, *broadcast_storm*, Δt_k , $layer_2$, {<s_1, 1,3>, <s_2, 4,0>}. Observa-se que, quanto maior o peso, maior a probabilidade da ocorrência desta falha no segmento em questão. A propagação das falhas é um campo de incertezas, de forma que todo o raciocínio é feito em cima de probabilidades condicionais, classificando ainda os alarmes como candidatos ou significantes. Com base neste modelo, um ambiente de teste foi usado para determinação das probabilidades para cada alarme de acordo com as falhas. Constrói-se então seu domínio hierárquico de gerenciamento, onde os resultados inferidos são explicitados pela entrada:

{<d_l, s_m, F_z, w_m, Δt_k>},

que é enviada ao nível mais alto até o *back-end* do software que implementa o mecanismo. A variável d_l representa o domínio hierárquico definido. Ao isolar o componente em falha, o problema cai em um modelo CBR. Para melhor entendimento das relações de "causa e conseqüência" entre as falhas inferidas, técnicas de *data mining* vem sendo empregadas, buscando aumentar o espectro de abrangência do modelo para problemas mais complexos.

3.1.4 Raciocínio Baseado em Casos – CBR

Como uma evolução do RBR, no CBR a experiência passada é usada na solução de novos problemas, permitindo sua adaptação a novas situações. A aquisição do conhecimento é mais intuitiva que nos sistemas especialistas e demanda baixa manutenção. A despeito destas vantagens, ainda há questões a resolver. Uma delas é a representação de casos para um sistema particular, bem como a representação dos relacionamentos entre casos e partes diferentes de casos.

O CBR é uma alternativa ao RBR. Seus objetivos são: aprendizado através da experiência, durante sua execução; oferecimento de soluções para novos problemas, de acordo com experiências anteriormente ocorridas; e evitar a necessidade de manutenção. A idéia básica do CBR é o uso de uma biblioteca de casos de solução de problemas que possam ser reutilizados ou adaptados para novas situações.

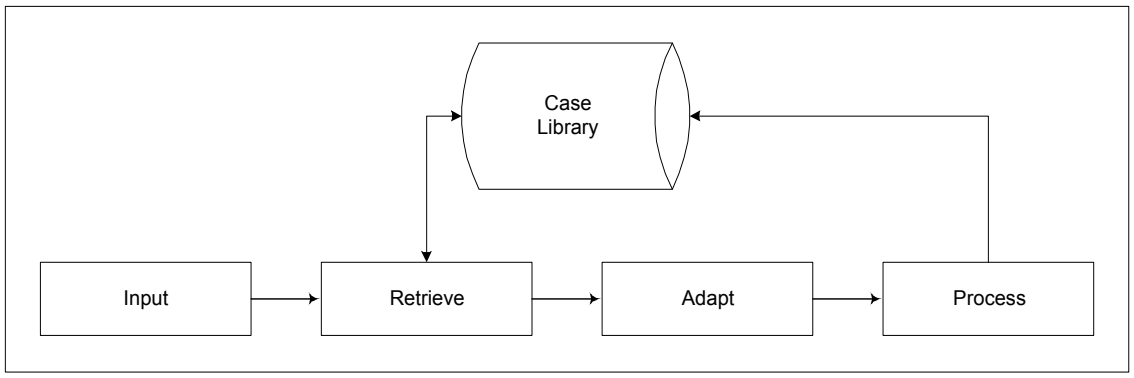


Figura 6 - A Arquitetura do CBR

Na figura 6 é apresentada a arquitetura empregada no modelo proposto em LEWIS (1993). Para sua implementação, usa-se um sistema de *trouble-ticketing*, onde é adicionado um módulo de CBR. Um *trouble ticket* nada mais é do que a descrição de um caso que contem informações sobre um determinado problema resolvido, sendo indicado como referência para determinadas situações onde o conhecimento de eventos envolvendo falhas já existe ou pode facilmente ser obtido.

O CBR desta aplicação emprega o conceito dos determinantes para acrescentar novos casos à biblioteca de *trouble-tickets*. Um exemplo de determinante seria o seguinte:

A solução para o problema "*file_transfer_throughput is slow*" é determinada de acordo com os parâmetros *network_load*, *bandwidth*, *packet_colision_rate* e *packet_deferment_rate*

Este determinante registra a informação de relevância entre classes de problemas na rede e conjuntos de atributos nos *tickets*. Essa relevância é usada para evitar-se a seleção de *tickets* similares, porém para problemas diferentes da falha em análise. Esses determinantes não funcionam como as regras no RBR, pois permitem uma adaptação do sistema com mudanças de configuração e de domínio. Novas situações são acrescentadas à base de dados, que é usada para solução de problemas. O processo de escolha das soluções (algoritmo CBR) envolve os seguintes passos: Segundo os determinantes, *trouble-tickets* são selecionados, e indexados de acordo com sua similaridade com o problema em pauta, sendo então seqüencialmente empregados na tentativa de solucioná-lo. A experiência com essas submissões é gravada na base, para futura referência, agregando a informação de sua eficácia, com o status "*good*" ou "*not good*". Se nenhum *ticket* possibilita a solução do problema, são sugeridas algumas

estratégias para adaptar uma solução existente semelhante à solução buscada. Uma delas é a adaptação parametrizada, onde valores de parâmetros são inferidos através de interpolações entre os parâmetros disponíveis. Supondo que o seguinte *ticket* está disponível:

```
trouble:            file_transfer_throughput = F
additional data:    none
resolution:         A = f(F), adjust_network_load = A
resolution status:  good
```

Supondo-se ainda que, numa falha semelhante, há uma saída do parâmetro *file_transfer_throughput = F'*, pode-se então propor para resolução ajustar o parâmetro *adjust_network_load para A'*, de acordo com a função descrita em *resolution*. A resolução para o problema poderá gerar então um novo *ticket* do tipo:

```
trouble:            file_transfer_throughput = F'
additional data:    none
resolution:         A' = f(F'), adjust_network_load = A'
resolution status:  good
```

A função f poderia ainda ser implementada através de inferência *fuzzy* ou uma árvore de decisão.

3.1.5 Outras abordagens

Além das abordagens apresentadas, podem-se encontrar combinações e variações destes modelos fundamentais. Um exemplo é o *Ripple-Down Rules* (RDR) YOSHIDA *et al.* (2002), cujo foco é a aquisição do conhecimento. Possui uma base binária, em forma de árvore. Cada nó possui dois ramos, onde cada um corresponde à satisfação ou não da regra. Uma vez que a inferência é feita em processo *top-down*, a última regra "disparada" num caminho particular dá a conclusão para o caso. Caso a conclusão não esteja correta, uma nova regra é adicionada, mas nenhuma regra é retratada ou corrigida. Se a solução para um caso não é encontrada, num caminho particular, conhecimento é adicionado àquele caminho, sem busca por outros caminhos. Se por um lado isso melhora a performance da solução dos problemas, por outro lado introduz repetição na base de conhecimento. Com o conhecimento da base teórica usada para a construção dos principais modelos arquiteturais para tratamento de falhas em redes, foram definidos os objetivos a serem alcançados através deste trabalho, que são descritos na próxima seção.

3.2 Sistemas de Diagnóstico de Redes baseados em CBR

Esta seção descreve aspectos relevantes de algumas propostas disponíveis na literatura, baseadas em CBR.

3.2.1 NetTrac

Trata-se de um sistema para controle de tráfego numa rede telefônica, feito através da alocação de recursos de acordo com a demanda (MELCHIORS, 1999). As informações de gerenciamento, disponíveis na rede, são capturadas pelo sistema. Estas informações são comparadas com os parâmetros disponíveis em situações armazenadas, num procedimento denominado de pré-processamento. Neste procedimento, tenta-se estabelecer uma similaridade a um quadro de problema disponível na forma de um *Problem Statement (PS)*. Caso encontre uma situação semelhante, o pré-processador envia os dados a um módulo monitor para a implementação de ações corretivas, propondo ajustes ou modificações quando necessário. Se não houver semelhança entre o caso atual e os PS disponíveis, este PS é transferido para um módulo Indexador/casador, que recupera casos de acordo com similaridades, podendo também ser modificado. Os atributos usados na indexação dos casos possuem relevância variável, de acordo com o domínio do problema, devido a grande diversidade de situações possíveis. O Criticador verificará a real aplicabilidade da solução proposta, à luz de possíveis danos na infra-estrutura. O usuário interage então com o sistema, aceitando ou não o tratamento proposto através de comandos de controle nos comutadores da rede. A figura 7 apresenta a arquitetura do sistema NETTRAC.

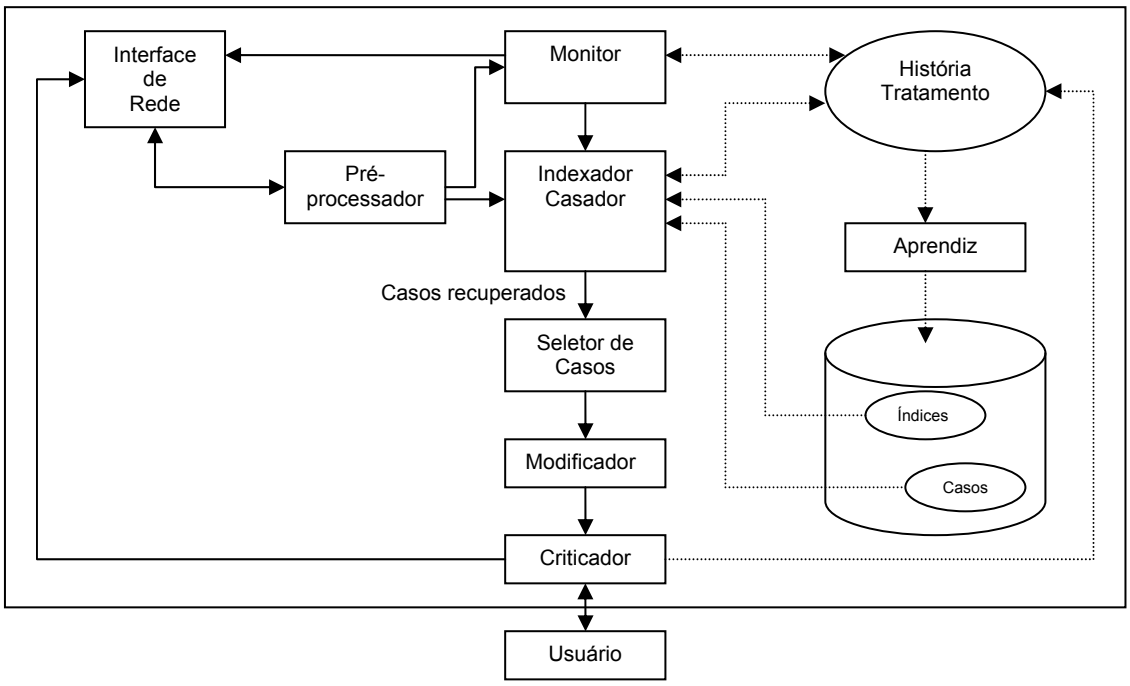


Figura 7 - Arquitetura do NETTRAC

Na figura 7, as linhas cheias representam o fluxo de informações de controle e de dados, e as tracejadas o fluxo de dados.

3.2.2 ExSim

O sistema ExSim (MELCHIORS, 1999), prototipado inteiramente em ambiente de simulação, destina-se ao gerenciamento de uma infra-estrutura de roteamento de redes WAN (*Wide Area Network*). O roteamento do modelo é estático, ou seja, há uma tabela de caminhos por onde os pacotes devem ser enviados de acordo com o endereço de destino. Tais caminhos podem estar momentaneamente congestionados ou permanentemente indisponíveis, gerando congestionamentos. Neste protótipo, um módulo "Simulador" gera situações de congestionamento com o objetivo de provocar mudanças de estado capazes de evidenciar sintomas de falhas através de alarmes, iniciando o processo de busca de uma solução para o problema.. A avaliação da aplicabilidade ou não de uma solução obtida na base de casos é exclusiva do sistema, não havendo interfaces com o usuário. Os atributos de casos consistem na combinação de informações que os caracterizam. A descrição do problema é realizada através de um atributo que consiste nas tabelas de roteamento, a informação da carga em todos os enlaces, a tabela da topologia da rede e o estado dos *gateways*. O atributo "tabela de roteamento", por sua vez, é representado por um conjunto de matrizes inteiras com os

endereços e a carga da rede por um conjunto de números positivos em ponto flutuante. As similaridades são calculadas pela comparação entre os componentes de cada atributo presente em ambos os casos, o atual e o armazenado. A figura 8 apresenta a arquitetura do ExSim.

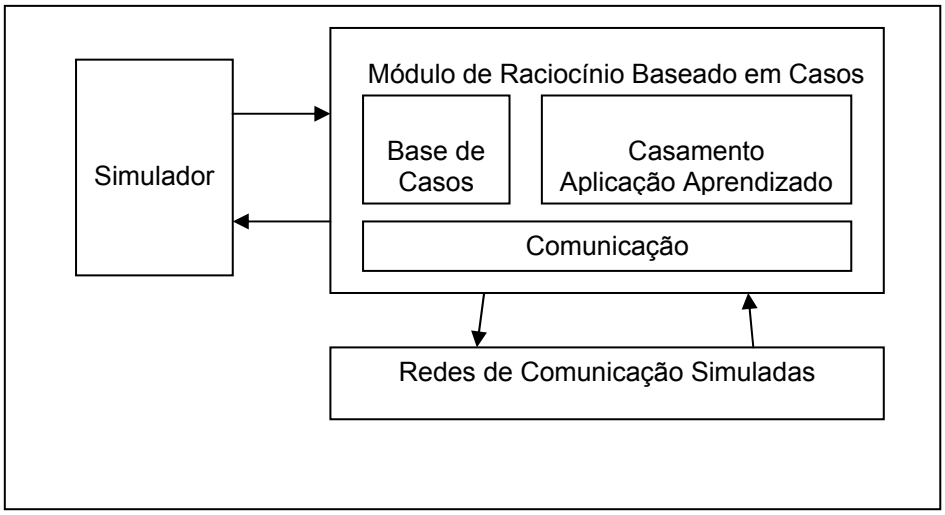


Figura 8 - Arquitetura ExSim

3.2.3 CRITTER

Sistema com objetivos bem semelhantes aos propostos neste trabalho, tem as seguintes características principais (MELCHIORS, 1999):

- O sistema CBR opera sobre um banco de registro de problemas observados, acompanhados de parâmetros capturados no momento da ocorrência do problema (sintomas);
- Define uma linguagem estruturada para os atributos relevantes através de palavras-chave, como por exemplo:
    - o atributo "*trouble*" pode ser preenchido com valores como "*file_transfer_througput = slow*"
    - o atributo "*additional data*" pode ser valorado com "*network_load= 20, collision_rate=15, user=31*"

Esta opção na construção do sistema reduz consideravelmente a complexidade necessária para o tratamento da linguagem natural tipicamente usada para caracterizar problemas de rede. Por outro lado, também reduz o escopo do aprendizado, já que impõe um conjunto finito de valores possíveis para os atributos, que por sua vez

também são previamente definidos. No entanto, tal limitação não chega a representar um grave problema na área de gerenciamento de redes, pois os sintomas, apesar de serem vários e poderem ser combinados e/ou propagados, possuem um domínio finito de possibilidades, uma vez definidos limites aceitáveis de parâmetros de monitoração de acordo com a topologia de cada rede.

A base de registro é complementada com um conjunto de informações sobre a relevância de um ou outro atributo, de acordo com o problema (sintoma) detectado. No CRITTER, esse conhecimento adicional é denominado "determinador", e possui a seguinte forma (exemplo): "*A solução para o problema – vazão da transferência de arquivos está lenta – é determinada observando a largura de banda, a carga da rede, a taxa de colisões e a taxa de retardo dos pacotes*". Estes determinadores operam como um conjunto de regras que podem ser refinadas automaticamente com o aprendizado do sistema.

Com os dados do caso atual e as informações adicionais indicadas pelos determinadores, busca-se o registro com maior similaridade, que é proposto ao usuário. Possui também uma estratégia de adaptação, que pode variar de acordo com a situação. Implementa, por exemplo, a adaptação parametrizada, quando há uma função de relação disponível entre parâmetros cujos rótulos estejam presentes tanto na solução do caso recuperado quanto no problema proposto, diferindo apenas pelo valor do atributo relevante para caracterização do problema. Por exemplo, se para um valor de *vazão_transferência_arquivo = F*, a solução implica no ajuste da carga de rede (parâmetro *ajustar_carga_rede*) é obtido pela função *A=f(F)*, então, num caso caracterizado pelo mesmo atributo relevante *vazão_transferência_arquivo = F*, a valoração referente ao ajuste da carga de rede *A* pode ser obtido (adaptado) através da função disponível no caso recuperado. Há outros métodos de adaptação, como por exemplo, a combinação de duas soluções possíveis para um mesmo tipo de problema, se a aplicação de uma solução proposta não resolve o problema e a segunda sim, um novo caso é armazenado com esse histórico. Outro tipo interessante de adaptação é originada na crítica feita por um especialista na solução proposta antes da aplicação da mesma, originando um novo registro de caso. Essa estratégia é especialmente interessante durante a carga inicial do sistema CBR. A arquitetura do sistema CRITTER é apresentada na figura 9.

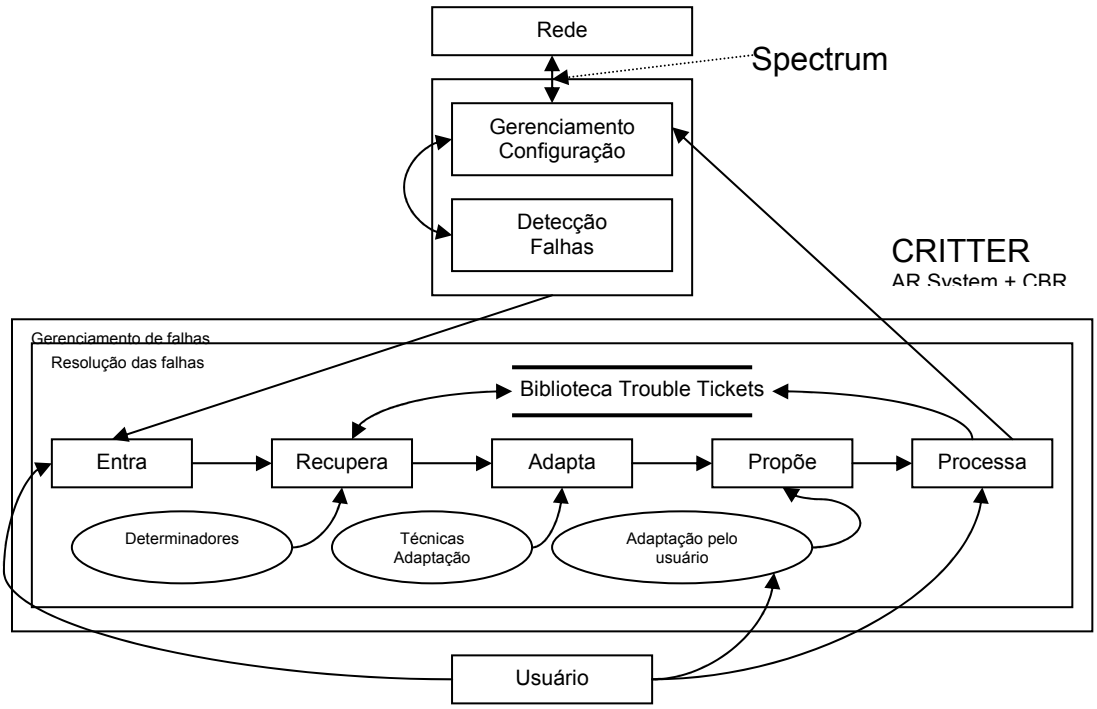


Figura 9 - Arquitetura do Sistema CRITTER

Na figura 9, podem ser destacadas as seguintes funcionalidades do sistema para cada módulo: O módulo entra recebe as situações de falha, oriundas de um outro sistema, chamado SPECTRUM, ou então do próprio usuário. O sistema AR provê as funções de gerenciamento das falhas, e o CBR é responsável pela resolução das falhas detectadas.

3.2.4 DUMBO

O sistema Dumbo (MELCHIORS, 1999), é uma das implementações de CBR com documentação disponível. Além disso, foi idealizado com uma finalidade semelhante à deste trabalho, que é a de migrar de um sistema de registro de eventos já disponível para um CBR. Por essas razões, o DUMBO foi usado como base para as decisões de projeto da implementação a realizar neste trabalho. Seu ponto de partida foi um sistema de registro de problemas chamado CINEMA, usado pelo centro de gerência de redes do POP-RS (*point-of presence* do Rio Grande do Sul). Sua finalidade era semelhante à do sistema legado "SOS" do IPqM, ou seja, meramente registrar ocorrências de forma textual, com o objetivo de permitir a criação de uma memória de casos passível de ser usada por qualquer usuário, ao surgimento de uma nova falha. Não permite, no entanto, a busca indexada por similaridades entre as falhas, orientada pelos sintomas, o que torna a busca exaustiva e complexa com o aumento do número de casos. Usou também informações disponíveis em manuais de *troubleshooting* de redes,

como por exemplo: sintomas como *alta taxa de erros de saída e de entrada* indicam como prováveis causas *má conexão física ou outros problemas físicos na rede, sem perda de conectividade*, mas um sintoma isolado como *alta taxa de erros apenas na entrada* indica como provável causa a *saturação da estação local*.

Na construção dos registros dos casos, optou-se pelo uso exclusivo de uma linguagem descritiva formal e bem estruturada, devido à complexidade do uso de textos livres para o cálculo de similaridades, permitindo-se, no entanto, a inserção de um campo adicional neste formato livre apenas para complementar o registro, de maneira informativa. Os principais tipos de problemas são então listados previamente, permitindo-se, porém, a inclusão de novas ocorrências de problemas, para inclusão na lista. As informações relevantes para cada tipo de problema são também definidas, visando uma possível complementação do caso proposto antes da busca no CBR. O DUMBO não implementa a captação de problemas diagnosticados automaticamente, apenas por inserção manual por especialistas. Neste trabalho, propõe-se que, com os modelos de falhas descritos através das RP, associados às funcionalidades do módulo *probe*, essa tarefa possa ser realizada, além da pesquisa manual, da mesma forma que no DUMBO. Uma observação bastante importante feita pelos autores do DUMBO é que os problemas em redes tipicamente são diagnosticados inicialmente de uma forma, porém após investigação mais detalhada revela-se terem sido originados por outra classe de problemas. Verificou-se também que uma mesma falha pode ocasionar tipos de problemas completamente diversos e em vários níveis. Uma placa de rede inoperante, por exemplo, pode resultar num problema como ”*aplicação xxx não acessa o BD*” ou então “*não é possível acessar o servidor*”. Tais observações reforçam a necessidade de se mapear as principais falhas através de modelos que indiquem claramente quais atributos complementares devem ser obtidos para o correto diagnóstico.

Uma vez obtidas as características do problema e dos seus prováveis tipos, faz-se uma busca na biblioteca de casos. Os casos recuperados são classificados de acordo com o seu grau de similaridade, levando-se em conta diferentes graus de relevância dos atributos para o caso descrito, o que provoca a definição de diferentes pesos no cálculo da similaridade. A indexação é feita de acordo com essa similaridade. No processo de recuperação é possível o refino interativo para a busca dos casos, com a introdução manual de atributos solicitados ao usuário pelo sistema.

Os problemas são representados através de redes semânticas, uma vez que tal representação permite a criação de uma hierarquia dos tipos de problemas útil para a

implementação de pesos diferenciados no cálculo da similaridade. Essa representação gráfica do conhecimento é composta por nodos e arcos que os relacionam, conforme o extrato da rede semântica dos tipos de problemas apresentado na figura 10.

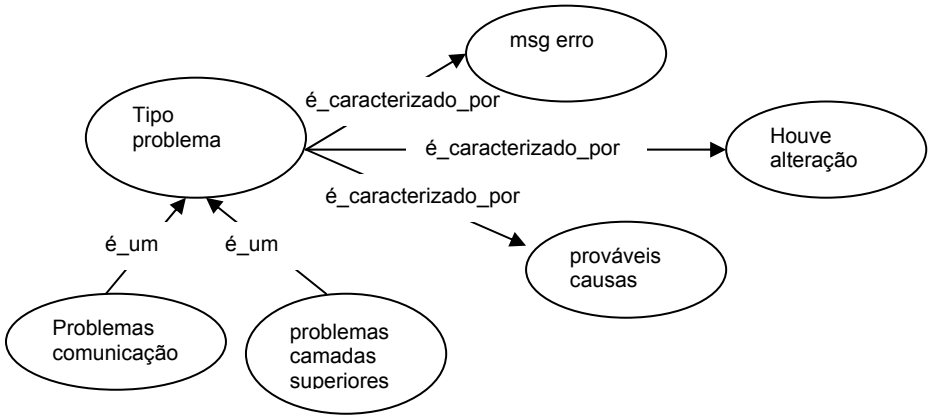


Figura 10 - Extrato da Rede Semântica dos tipos de problemas do DUMBO

A rede semântica é ampliada com o detalhamento dos problemas de comunicação e nas camadas superiores. Esta representação lógica concentra o conhecimento especialista acerca dos sintomas e os respectivos problemas típicos. A arquitetura do sistema DUMBO é apresentada na figura 11.

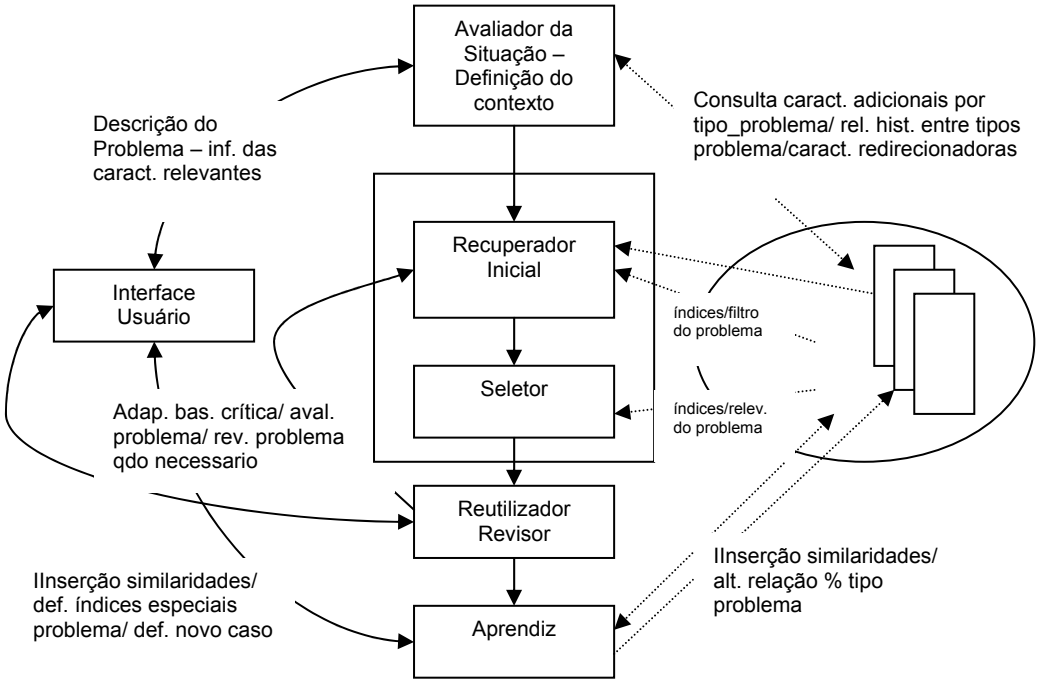


Figura 11 - Arquitetura do Sistema Dumbo

Na figura 11, podem ser destacadas as seguintes funcionalidades: O módulo *Avaliador da Situação/Definição do Contexto* é responsável pela obtenção das informações para a descrição do problema. Informações não disponíveis são solicitadas ao usuário. Uma vez identificado o tipo de problema provável e, através de regras redirecionadoras, se outros tipos de problemas devem ser consultados, o módulo *Recuperador Inicial* busca os casos na biblioteca. O módulo Seletor classifica os casos recuperados segundo as similaridades e a relevância de suas características. O caso (ou casos) recuperado pode ser aceito pelo usuário, ou então a situação ser refinada pelo módulo *Reutilizador/Revisor*, interagindo com o usuário, e retornando ao módulo *Recuperador Inicial* para nova busca. Se o sistema não for capaz de sugerir nenhum caso, o módulo *Aprendiz* armazena o mesmo para fins de gerenciamento, acrescentando ao mesmo a experiência ora adquirida com o novo caso.

Conforme já dito anteriormente, o sistema DUMBO implementa uma linguagem formal para descrição dos casos. Essa descrição representa as informações acerca do problema e do ambiente no qual o mesmo ocorreu. Uma vez que os registros de ocorrências de problemas tradicionais, em texto livre, omitem ou generalizam detalhes importantes para a atividade de catalogação de casos, o sistema define um conjunto de informações relevantes para cada tipo de problema, a partir da declaração, por parte do usuário, de que se trata de um tipo específico de problema. Se, no entanto, se buscasse a total cobertura de todos os casos possíveis a partir de um conjunto de informações, seria exaustivo, devido à complexidade dos problemas e da possibilidade de alarmes propagados identificarem incorretamente um determinado tipo de problema. Assim, o sistema DUMBO implementa um processo de refino dos casos, onde, após a primeira recuperação, são identificadas novas informações relevantes, provenientes dos casos inicialmente identificados como similares, apenas para o caso corrente. Essas informações, denominadas de características específicas, são solicitadas ao usuário para o processo de refino.

Os casos, cadastrados hierarquicamente pelo tipo de problema, são acessados através de índices. Esses índices, peça-chave na performance do sistema, são formados por características importantes que definem os casos. No sistema DUMBO, os casos foram indexados pelo tipo de problema. Toda a introdução dos mesmos no sistema é feita de maneira manual, com a identificação das informações que poderiam contribuir para a identificação de casos similares. A obtenção de tais informações é feita de forma manual pelo próprio usuário. A recuperação dos casos similares é realizada através de

regras especialistas pré-definidas, e na relação histórica entre tipos de problema. As regras especialistas, denominadas regras *redirecionadoras*, atribuem pesos para cada teste de tipo de problema, conforme exemplo a seguir:

```
      SE problema persiste utilizando IP = ´não´
      ENTAO tipo de problema ´serviços-resolução de nomes´ PESO 3

      SE falta de acesso = ´constante´ ou falta de acesso =
´intermitente´ E
            tipo de problema não é ´conectividade-genérico´ E
            tipo de problema não é ´conectividade-físico e config/HW E
            tipo de problema não é ´roteamento/endereçamento´
      ENTAO tipo de problema ´conectividade-genérico´ PESO 3
```

A função destas regras é obter tipos de problemas que poderiam também ser escolhidos em outras situações, embora não possuam os sintomas principais. Com relação às relações históricas, o sistema implementa uma tabela onde é armazenado para cada tipo de problema o percentual de vezes que o problema foi, no passado, causado por outros tipos. A probabilidade final, então, é calculada pela equação 1.

Equação 1 - Cálculo da probabilidade para tipo de problema

$$prob\ (T, C) = \frac{pred\ (T, C) + I * phist\ (T, C)}{I + 1}$$

onde:
T: Tipo de problema
C: Caso corrente
pred(T,C): probabilidade segundo regras direcionadoras
phist(T,C): probabilidade segundo relação histórica
I: importância da relação histórica

O sistema DUMBO utiliza o método da recuperação orientada a índices (VON WANGENHEIN, 2003). Na fase de pré-processamento, todos os casos são analisados e indexados segundo o tipo de problema provável, à luz das similaridades e relevâncias de cada característica (sintoma) do problema. Na fase seguinte, a recuperação é feita através da estrutura de índices gerada e os casos mais similares são determinados. Tal método confere grande performance ao sistema, por reduzir consideravelmente o custo computacional do processo de recuperação, porém possui as seguintes desvantagens:

- Impossibilita consultas *ad-hoc*, uma vez que demanda a criação de um índice para cada consulta;
- Alto custo decorrente de:
    - Geração e manutenção da estrutura de índices;
    - Requisição de espaço físico adicional para armazenamento de índices;

- Demanda grande flexibilidade do BD utilizado;
- Maior esforço de implementação, pois a forma de armazenamento dos casos, a estrutura e os tipos de consulta permitidos devem ser criteriosamente definidos à priori para permitir a criação de algoritmos eficazes de indexação e de recuperação.

Uma vez obtido o índice por casos prováveis, a busca dos casos na base então é feita através do seguinte algoritmo (figura 12):

```
nodo_atual = raiz
para cada nodo folha tipo_de_problema repetir
    se tipo_de_problema ∈ tipos_problema_prováveis_selecionados
            nodo_atual = nodo folha
        se nodos folhas nodo_atual são casos
            selecionar casos
        senão
            nodo_atual=nodo folha cujo índice arco seja melhor
casamento
            selecionar  casos representados pelos folhas do nodo
atual
```

Figura 12 - Algoritmo para recuperação de casos

Tais casos então devem ser classificados para apresentação ao usuário. Para isso, é usada importância relativa para cada tipo de característica, que é classificada com graus que variam de 5 a 0, de acordo com a sua importância, que vai de "essencial" até "não importante". Para os graus de relevância, foi criada uma tabela onde são definidos pesos para cada característica, de acordo com o tipo de problema. Nesta tabela, por exemplo, o peso da característica *há muitos erros* tem peso 5 para *performance*, mas tem peso 3 para *alto tráfego*. A similaridade entre o tipo de problema com o caso da base é feita através da função de probabilidade da equação 1. A similaridade entre casos (proposto e da base) é feita através do somatório da similaridade entre cada característica multiplicada pela relevância desta para o contexto do caso recuperado. Para distinguir-se entre a existência ou não de um valor para comparação de similaridade, foi criado o fator *Confiança* de cada característica, com valor 1 para presente e 0 para ausente. Além disso, é introduzido o fator *Confiabilidade*, que em função da *confiança* e dos pesos de cada característica, busca espelhar o quanto a similaridade entre os dois casos é confiável. Por fim, o ordenamento é realizado com base na similaridade e na confiabilidade, sendo que o primeiro fator ainda é ponderado com a importância de similaridade frente à *confiabilidade*. Tal valor foi arbitrado em 2. As equações 2, 3 e 4 são utilizadas para o ordenamento dos casos.

Equação 2 - Função de Similaridade

$$Similaridade\ (Cr,R) = \frac{\sum_{i=1}^{n} Wi\ *\ sim\left(f_i^c, f_i^R\right) * Ci}{\sum_{i=1}^{n} Wi\ *\ Ci}$$

Equação 3 - Função de Confiabilidade

$$Confiabilidade\ (Cr,R) = \frac{\sum_{i=1}^{n} Wi * Ci}{\sum_{i=1}^{n} Wi}$$

Equação 4 - Fator de Ordenamento

$$FatorOrdenamento\,(R) = \frac{I\ *\ Similaridade(Cr,R) + Confiabilidade(Cr,R)}{I+1}$$

$Onde: Wi - importância\ \ da\ característica\ \ i$

$f_i^C\ e\ f_i^R - valores\ das\ características\ f_i\ no\ caso\ corrente\ e\ no\ selecionado$

$Ci - confiança\ na\ similaridade\ (1\ presente, 0\ ausente)$

$I - importância\ da\ \ similaridade\ \ frente\ a\ \ confiabilidade\ (atual\ \ 2)$

Após a seleção dos melhores casos seu emprego ainda pode ser precedido por uma etapa de refino, com a introdução interativa de novos parâmetros para uma seleção mais apurada. Uma vez escolhida a melhor solução, o caso é encerrado com duas situações possíveis. Caso a solução seja satisfatória, não há nada a aprender. Se, no entanto, a solução não foi satisfatória, dependendo de uma adaptação para seu emprego, tal experiência é acrescentada ao caso e o mesmo é integrado à base como um novo caso.

3.3 Escolha do Raciocínio Baseado em Casos

Nas tarefas de manutenção de Redes de Computadores, invariavelmente os problemas são resolvidos através de conhecimento especialista dos gerentes de redes. Os Sistemas Operacionais, apesar de gerarem um conjunto de alarmes que permitem uma análise primária da situação de falha, elevam o nível de abstração a um patamar que dificulta para o usuário a verificação objetiva dos sintomas. Essa abstração impede diagnósticos triviais. Ao ligar um computador ligado em rede, por exemplo, se o cabo estiver desconectado, poderá apresentar apenas a mensagem: "*não foi possível encontrar um servidor para o domínio xxx*", em ambiente Windows 2000. A resposta a

um simples *ping* no nó imediatamente mais próximo do nó em questão já permitiria, no mínimo, a sugestão para que o usuário verificasse o cabo ou a placa de rede, sem a necessidade de testes complementares. Outro aspecto é que estes especialistas resolvem em curto prazo casos "similares" a outros pelos quais ele já tenha passado. Se o especialista ainda não enfrentou situação igual a atual, inicia procedimentos sem metodologia, chamado tipicamente de "tentativa e erro". Naturalmente, técnicos de suporte mais capazes precedem estas tentativas por testes que permitam a redução do escopo de possibilidades para a razão do problema em curso. A maioria dos bons Gerentes de Redes inicia a exercer essa função com pouca experiência e, conforme vão enfrentando novos casos e os solucionando, têm o hábito de registrar as ocorrências de casos documentalmente ou mentalmente, indexando por sistema ou organizando a sua base de conhecimento de acordo com os sintomas. Tipicamente, é usada a forma de expressão conversacional tradicional, associada com eventuais mensagens de erro típicas dos Sistemas Operacionais e parâmetros observados.

Toda essa semelhança entre a administração típica de falhas em redes e a organização proposta pela metodologia CBR já justifica naturalmente o seu uso para a criação de um sistema especialista de assessoramento no diagnóstico de falhas em Redes de Computadores.

3.4 Especificação do Protótipo de Diagnóstico

Muitas das falhas ocorridas são facilmente modeláveis em grupos, uma vez que a comunicação nas redes só é possível devido ao uso de protocolos de comunicação padronizados e estáveis. Há propostas na literatura que constroem sistemas para diagnóstico de falhas em redes com base em modelos, como as Redes de Petri (ROUVELLOU, HART, 1995, AGHARSARYAN *et al.,* 1997, USHIO *et al.,* 1998). Neste tipo de abordagem, as transições de um estado normal para um estado de falha ocorrem devido ao estouro de *time-outs* pré-configurados e disponíveis, ou através de alarmes recebidos. Muitas falhas, no entanto, são percebidas através de sintomas nem sempre determinísticos da causa real, por se tratar de uma manifestação em cascata. A figura 13 ilustra um exemplo deste tipo de situação.

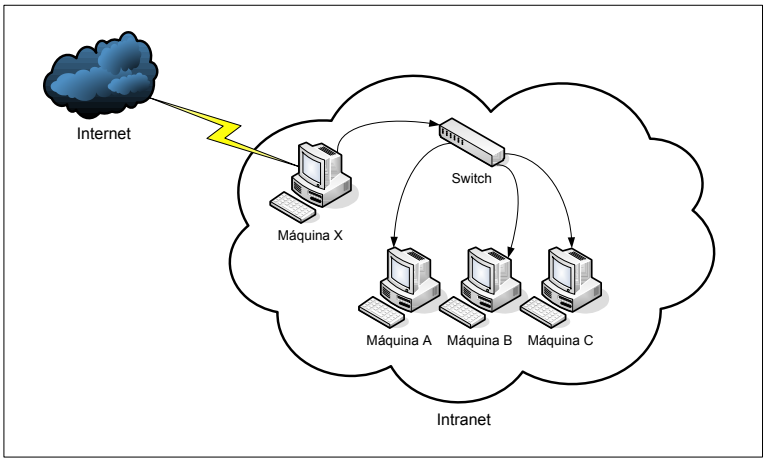


Figura 13 - Cenário de operação de um worm

Na figura 13, a máquina X é contaminada por um *worm*, cuja atuação típica é a de propagar-se em "segundo-plano", de forma silenciosa e independente de ações do usuário, utilizando-se apenas da infra-estrutura de rede e de vulnerabilidades específicas dos Sistemas Operacionais. As máquinas A, B e C relatarão "rede lenta", e um analista de suporte poderá, no seu processo normal de investigação, constatar a poluição da rede por *broadcasts* ARP e saturação do barramento. Na verdade, tais sinais são decorrentes do comportamento típico de um *worm*, que, após instalar-se, "descobre" o endereço da máquina contaminada e busca abrir conexões de transporte através de vulnerabilidades do Sistema Operacional com as máquinas pertencentes ao mesmo escopo de rede. Para isso, a Estação contaminada passa a gerar mensagens ARP, enviadas em *broadcast*, inquirindo o endereço de enlace (MAC) de cada possível máquina da rede através de seu endereço de rede (IP). Desta forma, ao se atender um usuário de máquinas na situação A, B e C, muito provavelmente observará o alto *broadcast* e a sobrecarga da rede, tendendo a dar um diagnóstico de excesso de máquinas no barramento. Este diagnóstico indicaria como solução a segmentação da rede, algo que atingiria os sintomas, reduzindo a sensação de lentidão, mas não a causa do problema. Com o sucesso na proliferação do *worm*, os sintomas voltariam a se manifestar.

A proposta deste trabalho é usar ferramentas de auxílio à obtenção de sinais, que pudessem ser incorporados aos Sistemas Operacionais. Uma vez que a falha possua similaridades a um modelo disponível, pode-se obter vantagens em relação ao CBR tradicional, já que um subconjunto de casos menor do que o original poderá ser pesquisado, independentemente da estratégia de recuperação utilizada. Espera-se, com isso, que a busca seja otimizada. Poder-se-ia ainda agregar aos modelos a variável

temporal, bem como lógica *Fuzzy* (PEDRYCZ, CAMARGO, 2003), para permitir uma comparação entre os mesmos em situações reais de falhas.

A análise das propostas disponíveis na literatura indica que a principal deficiência se encontra no início da operação de diagnóstico: a definição dos sintomas e sinais. Como os casos são mais ou menos similares em função da existência ou não destes elementos, a não percepção de um ou mais parâmetros típicos de falha naturalmente conduzirá a resultados afastados da situação real. Os mecanismos de "adaptar" e de "completar" o caso reaproximam casos onde tais parâmetros estejam indisponíveis, mas aumenta o número de situações prováveis. Com isso, reduz-se a eficiência na solução das falhas. O objetivo maior deste trabalho é, então, atuar na primeira fase do problema: a sua percepção, e com isso buscar maior eficiência na resolução dos mesmos.

São elementos Básicos de um Sistema CBR:

- Representação do Conhecimento (casos)
- Medidas de Similaridade
- Adaptação de Casos
- Aprendizado

O ciclo do CBR envolve as seguintes tarefas principais:

- Recuperar o caso mais similar ao atual
- Reutilizar esse caso para tentar resolver o problema
- Revisar a solução proposta à luz do resultado de sua implementação
- Reter a experiência representando o caso atual (no todo ou em partes) pra reutilização futura

Aspectos inicialmente definidos:

- Serão usados casos já registrados no ambiente de teste disponível. Uma vez que não havia metodologia de registro voltada para a percepção do caso, e sim para as soluções, os mesmos deverão ser manualmente adaptados para maior utilidade;
- O uso de elementos otimizadores, capazes de capturar parâmetros de falhas já foram descritos na literatura. Há propostas descrevendo o uso desta funcionalidade (*probe*), associada às regras descritas em linguagem formal, para diagnosticar falhas (BRODIE *et al.,* 2002). O enfoque neste trabalho é o de apenas utilizar o que já está disponível (via ICMP e SNMP) e passível de

incorporação nos Sistemas Operacionais, não como um aplicativo, mas sim como uma opção do mesmo, que pode ser ativado ou não. A arquitetura do trabalho proposto está ilustrada na figura 14.

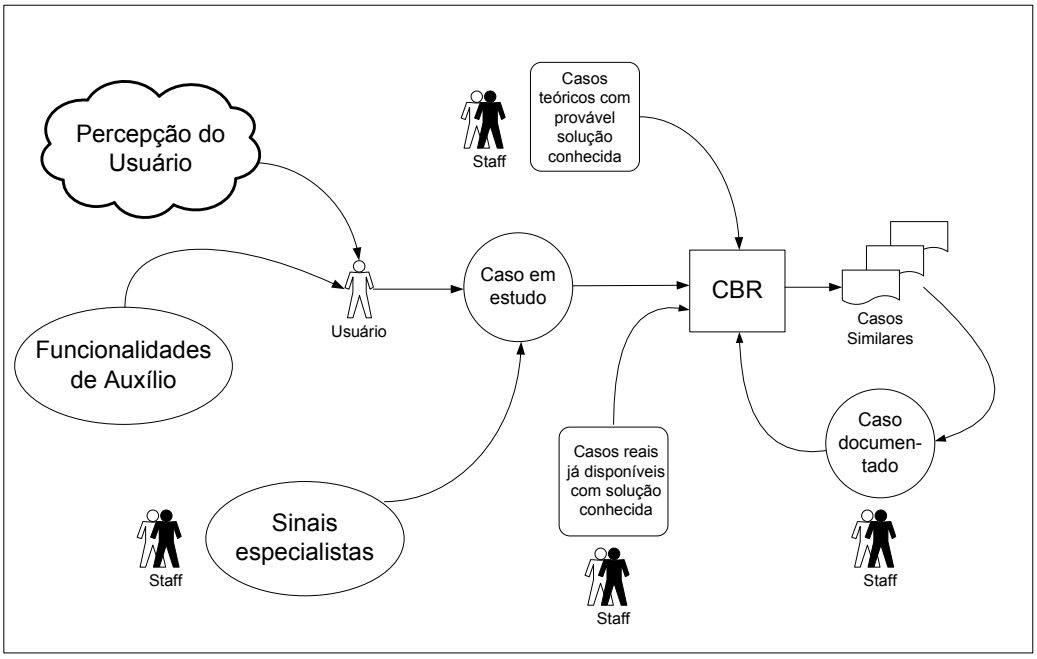


Figura 14 - Arquitetura Proposta

Na figura 14, são destacadas as funcionalidades principais da proposta deste trabalho: o usuário, ao perceber os primeiros sinais típicos de anormalidade (rede lenta, intermitente ou sem conexão), busca apoio com o *staff* de especialistas (típico *help-desk* da maioria das empresas). Para otimizar a resolução do problema, poderão ser agregadas informações de grande valor, os sinais especialistas, que são diferenciais para distinção entre tipos de falhas. Estes sinais serão discutidos no próximo capítulo. Estes sinais, capturados da rede, da máquina em análise ou dos dispositivos de conectividade, são considerados diferenciais por permitir a redução significativa de possibilidades de diagnóstico, facilitando assim a obtenção de maiores similaridades com casos mais prováveis de estarem sendo experimentados. São utilizadas ferramentas próprias de gerenciamento de redes, como *sniffers*, gerentes SNMP e comandos de linha, não sendo comum o seu uso por usuários finais, e sim pelo pessoal do *staff*. Para possibilitar que o usuário tenha uma percepção mais clara da ocorrência, são propostas neste trabalho a incorporação de funcionalidades de auxílio ao gerenciamento, especialmente para tratamento do protocolo ICMP. Este protocolo, criado especificamente para o tratamento de erros em redes no contexto da pilha de protocolos TCP/IP, não disponibiliza interfaces de simples manejo nos Sistemas Operacionais atuais. Uma

aplicação foi desenvolvida neste trabalho para este tratamento do ICMP de forma residente na memória das máquinas, sendo trivial a incorporação dessa facilidade como uma funcionalidade adicional, como já se faz com antivírus e *firewall*. Com isso, tem-se um caso a ser submetido ao sistema CBR. Neste trabalho, utiliza-se uma ferramenta bem conhecido na área de CBR, o *CBR-Works*, para o desenvolvimento da aplicação. A decisão de se utilizar o *CBR-Works* deveu-se ao grande número de trabalhos acadêmicos onde o mesmo é citado. Além disso, há todo um esforço de padronização de documentação própria para o CBR com a qual o *CBR-Works* é compatível, conforme será explanado no capítulo seguinte.

Uma vez obtidos os casos similares no *CBR-Works*, o *staff* avalia os diagnósticos sugeridos. São buscados sinais diferenciais adicionais, conforme o caso, e por final testa as soluções. Satisfatórios ou não, os resultados são documentados e lançados no sistema para enriquecimento da base.

Para materializar a viabilidade da proposta deste trabalho, o capítulo 7 descreve os resultados obtidos, considerando-se as estatísticas de resolução de problemas sob várias óticas diferentes.

Este capítulo descreveu os fundamentos principais deste trabalho, a proposta de implementação e os passos a serem seguidos. O próximo capítulo apresenta uma metodologia padronizada para documentação de sistemas CBR, que permitirá a continuidade do investimento realizado em pesquisa para elaboração deste trabalho.

4 A Metodologia INRECA II

O uso de uma metodologia própria para o desenvolvimento de aplicações CBR visa, em primeiro lugar, garantir um processo contínuo, em que os resultados das primeiras versões possam ser utilizados para refinamento do modelo, bem como a geração de uma documentação padronizada que permita a preservação do conhecimento obtido durante o desenvolvimento da aplicação CBR. Além disso, há outros benefícios, como o aumento da produtividade, qualidade do modelo e um padrão de comunicação entre desenvolvedores e clientes para melhor entendimento dos requisitos a serem atingidos. Com estas metas, objetiva-se que os produtos CBR possam ser poderosos instrumentos para o gerenciamento da Tomada de Decisões nas empresas.

A metodologia INRECA-II (*INduction and REasoning from CAses*) apresentada em BERGMANN *et al.* (1998) é baseada em duas áreas da engenharia de software, a abordagem *Experience Factory* (BASILI *et al.,* 1994, ALTHOFF *et al.,* 2005) e a Modelagem de Processos de Software (ROMBACH, VERLAGE, 1995).

4.1 Experience Factory

Essa técnica é motivada pela observação de que um negócio de sucesso requer a combinação de soluções não apenas técnicas, mas também gerenciais. Trata-se de um processo recursivo que suporta o aprendizado e o *feedback*. As tecnologias-chave para suporte a estes requisitos são: modelagem, métricas, reuso de processos, produtos e outras formas de conhecimento relevante para o negócio. A *Experience Factory* atua como repositório da experiência analisada e sintetizada, que é fornecida para os vários projetos sob demanda. A figura 15 apresenta um modelo da *Experience Factory* e seus macro-componentes.

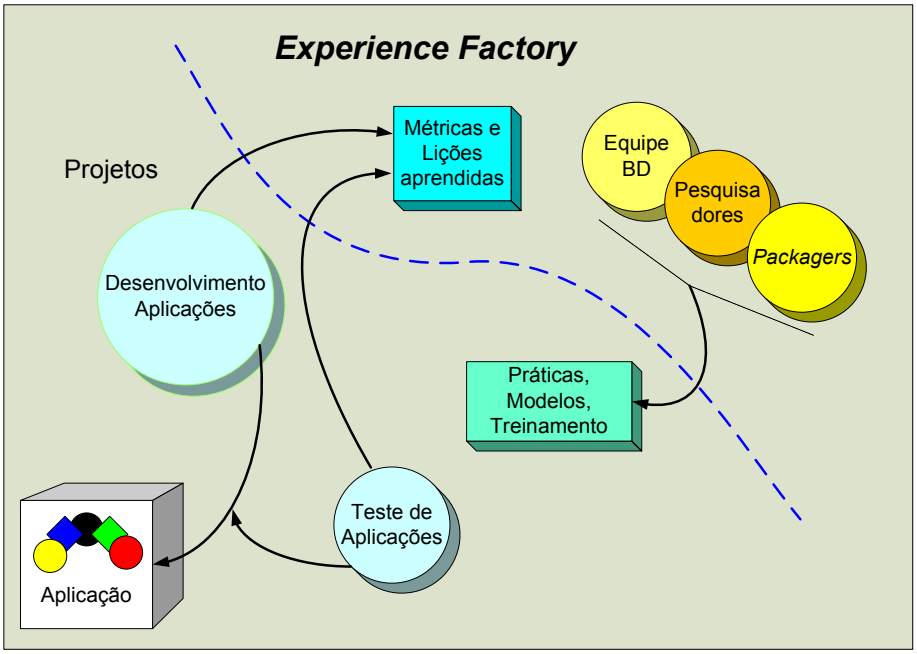


Figura 15 - Experience Factory

O *Staff* da *Experience Factory* armazena os dados usados durante o desenvolvimento e a operação dos seus produtos, analisa estes dados, sugere e conduz experimentos adicionais, e ao final agrupa na forma de um *package* esta experiência, contendo práticas, modelos e treinamento que pretendem contribuir para contínuo aprimoramento e obtenção de vantagens competitivas. Uma outra forma de ilustrar a técnica é apresentada na figura 16.

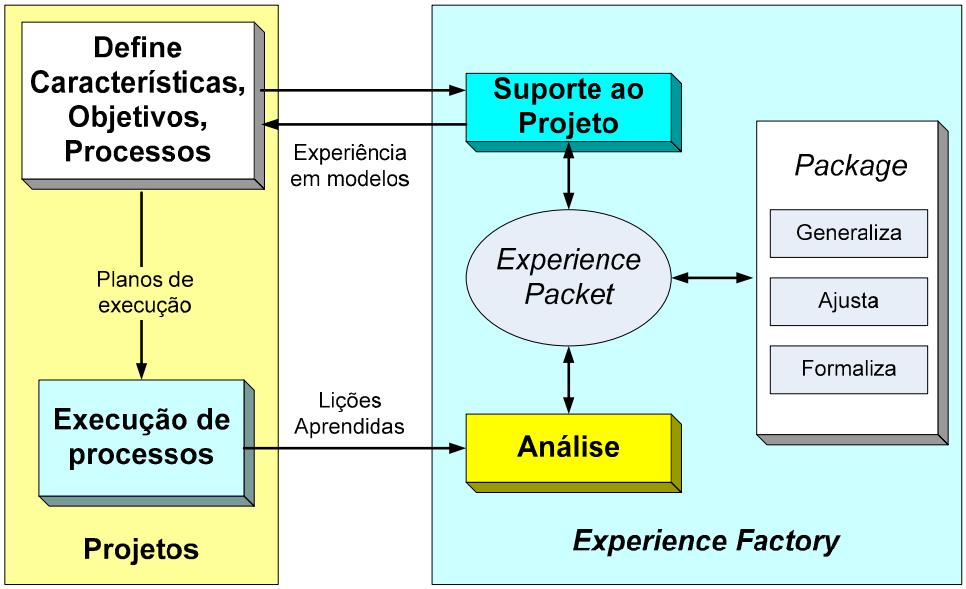


Figura 16 - Abordagem Experience Factory

Como se pode observar na figura 16, o mais importante produto de uma *experience factory* é o *experience packet*, cujo conteúdo e estrutura vão depender do tipo de experiência agrupada no *package*.

4.2 Modelagem de Processos de Software

Esta abordagem descreve a engenharia do produto, ou seja, do software a ser produzido. Abrange técnicas de engenharia de software, como:

- Engenharia de requisitos, design do sistema a ser desenvolvido, codificação e outros detalhes;
- Processos gerenciais, como documentação, gerenciamento do projeto, garantia de qualidade;
- Processos organizacionais, mais intrínsecos aos processos de negócio em si, nas interfaces com o sistema a ser desenvolvido.

Dentre as várias abordagens diferentes para a modelagem de processos, uma notação comum usa nomes particulares como *processos, métodos, produtos, objetivos (goals) e recursos*. Um processo é um passo único num projeto de desenvolvimento de software. Cada processo tem um objetivo (*goal*) próprio, e ele consome, produz ou modifica certos produtos, onde estão incluídos os sistemas (software executável), bem como sua documentação, referente ao seu *design* e manuais de usuário. Para um processo ser inicializado, um *método* apropriado deve ser indicado. Um exemplo de método simples pode ser uma descrição textual de como atingir o objetivo do processo. Um método mais complexo decomporia o processo num conjunto de sub-processos para, de forma colaborativa, atingir seus objetivos determinados.

Na metodologia INRECA-II, os modelos de processos de software são usados para representar a experiência de desenvolvimento dos CBR, cujo resultado será armazenado no *experience packet*. Processos de software, quando representados, podem ser bastante abstratos, representando apenas a definição do modelo do domínio, definição do modelo de similaridade e a aquisição dos casos. Mas pode também ser detalhado e específico para um projeto particular, como a análise de dados obtidos de uma aplicação de gerenciamento de rede (como um analisador de tráfego), selecionando os parâmetros relevantes, etc. por exemplo, no caso específico deste trabalho, de acordo com a filtragem prévia dos sintomas e dos dados coletados, selecionar os parâmetros representativos de situações de erro relevantes, e com isso obter grande refinamento da aplicação CBR. A Modelagem de Processos possibilita a construção de um modelo

hierárquico de processos, e essa propriedade é importante para a elaboração do *Experience Packet*.

4.3 A Estrutura do Experience Packet

O *Experience Packet* é organizado em três níveis hierárquicos de abstração: um Nível Genérico Comum, um *Cookbook-level* e um nível específico do projeto, conforme ilustrado na figura 17. O *Experience Packet*, para os objetivos deste trabalho, representa a modelagem da Base de Experiências.

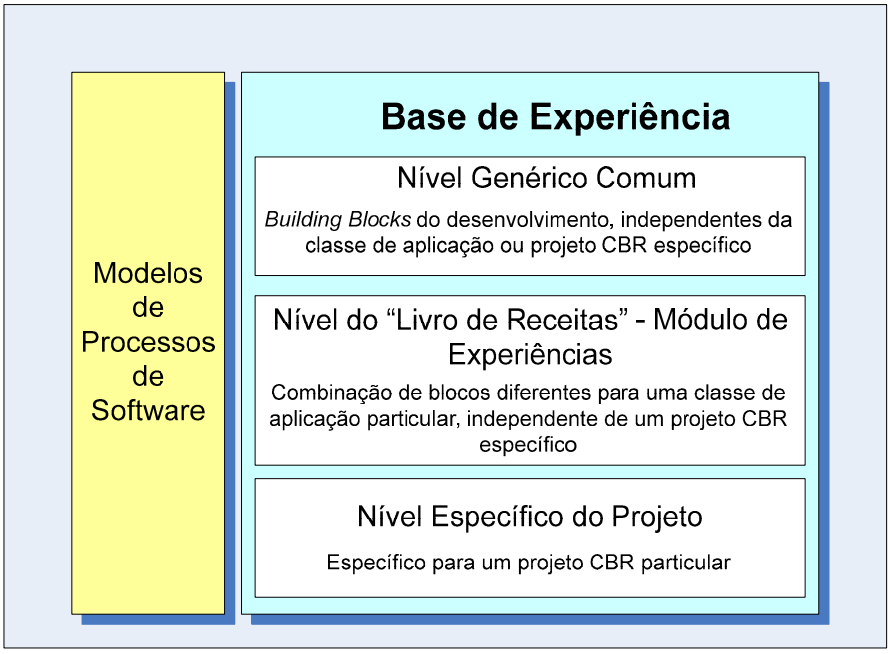


Figura 17 - Experience Packet

No Nível Genérico Comum, são descritos processos, produtos e métodos que são comuns a um determinado espectro de diferentes aplicações CBR. As descrições destes elementos constituem o fluxo de ações da metodologia. Um **Processo** genérico descreve uma classe de processos através da definição das seguintes propriedades (BERGMANN, ALTHOFF, 1998):

- Um objetivo particular tal como um passo especificando o que deve ser alcançado;
- Um conjunto de diferentes métodos alternativos que podem ser usados para implementar tal passo;
- Entradas, saídas e produtos modificados que descrevem quais produtos são requeridos no início, quais devem ser obtidos ao fim do processo e quais devem ser alterados durante o mesmo;

- Um conjunto de recursos (agentes ou ferramentas) que são requeridas para execução do passo, definidos através das necessárias especificações ou qualificações que o agente ou ferramenta deve possuir para ser usada no processo.

Métodos são elementos que contém uma detalhada especificação de uma forma particular de alcançar um determinado objetivo de um processo, podendo ser simples ou complexo. Métodos simples fornecem apenas uma descrição do que fazer para atingir o objetivo, e métodos complexos especificam um conjunto de sub-processos, um conjunto de produtos intermediários e um fluxo entre os sub-processos. Essa abordagem permite a definição de modelos de processos hierárquicos e flexíveis.

O principal objetivo do processo é criar ou modificar produtos. **Produtos** são componentes modelados através de Descrições Genéricas de Produto, que declaram propriedades que todos os produtos de um certo tipo devem possuir. Uma definição de domínio descrita na linguagem CASUEL (BERGMANN, 2001) é um exemplo de produto. Um produto pode ser decomposto em sub-produtos, através da definição de tipos de atributos, classes de objetos, casos e medidas de similaridade específicas.

Recursos são entidades necessárias para execução das tarefas, podendo ser agentes ou ferramentas. **Agentes** são modelos para pessoas ou grupos de pessoas (exemplos: gerentes, especialistas no domínio em discussão, *designers* ou programadores), que são designadas para executar um processo. A mais relevante propriedade de agentes é a sua qualificação. **Ferramentas** são usadas para suportar o estabelecimento de um processo e que pode ser descrito através de uma especificação (exemplos: ferramenta de modelagem, aplicação CBR, construtor de IHM). Usando as qualificações e especificações definidas no processo genérico, podem-se determinar os agentes e ferramentas necessárias para um processo específico.

No nível seguinte, o "livro de receitas", encontram-se os módulos de experiências. Processos, produtos e métodos são refinados para uma classe particular de aplicações, como por exemplo, o diagnóstico de falhas em redes de computadores. Para essa classe específica, o nível contém um módulo de experiência descrevendo como uma aplicação deve ser desenvolvida e/ou mantida. Esta descrição deve orientar o desenvolvimento da aplicação CBR. Todos os processos que sejam relevantes para uma determinada classe de aplicação devem ser conectados, formando um fluxo de produto para o projeto específico.

O nível inferior, específico do projeto, descreve a experiência no contexto de um único projeto em particular, que já tenha sido realizado. Contém informação como processos já realizados, custo dos mesmos, produtos já produzidos e métodos selecionados para executas os processos, pessoas envolvidas na execução de um processo em particular. É uma documentação completa do projeto que visa a obtenção da conformidade com requisitos de qualidade. A figura 18 ilustra o detalhamento descrito acima.

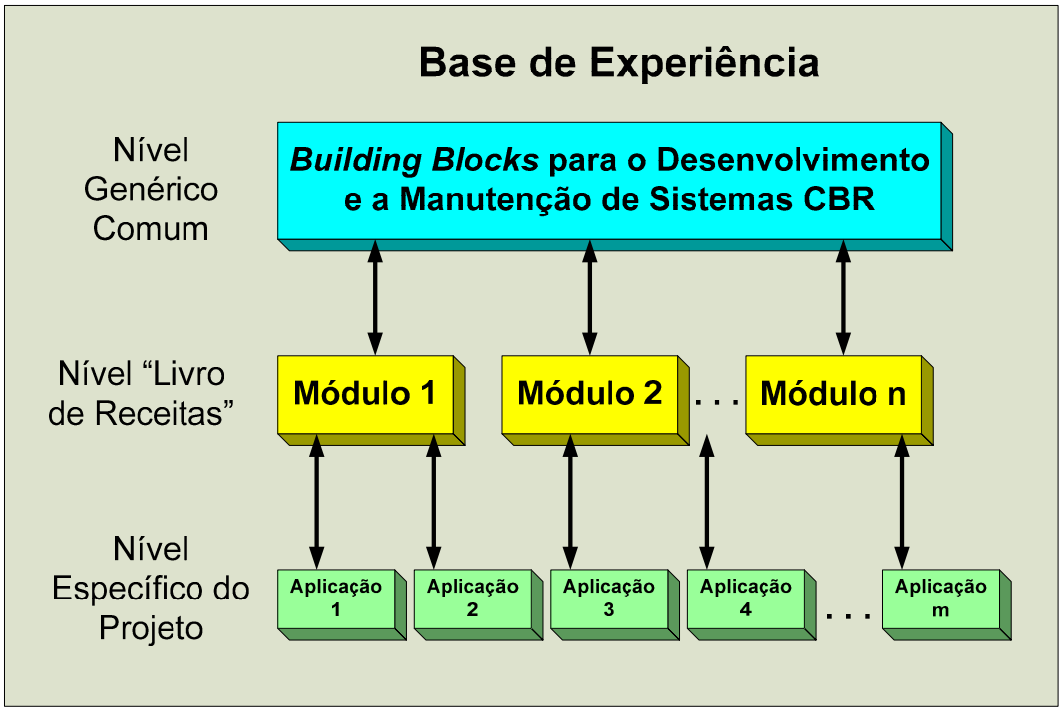


Figura 18 - Base de Experiências - Detalhamento

4.4 Documentação da Base de Experiências

A documentação dos processos, produtos, métodos, agentes e ferramentas, armazenados numa base de experiências são documentados na forma de *fichas*. Uma ficha é um formulário formatado para documentar um item em particular. Contém vários campos pré-definidos a serem preenchidos, além de *links* para outras fichas. Um processo genérico, por exemplo, pode ser descrito através de quatro fichas conforme ilustrado na figura 19.

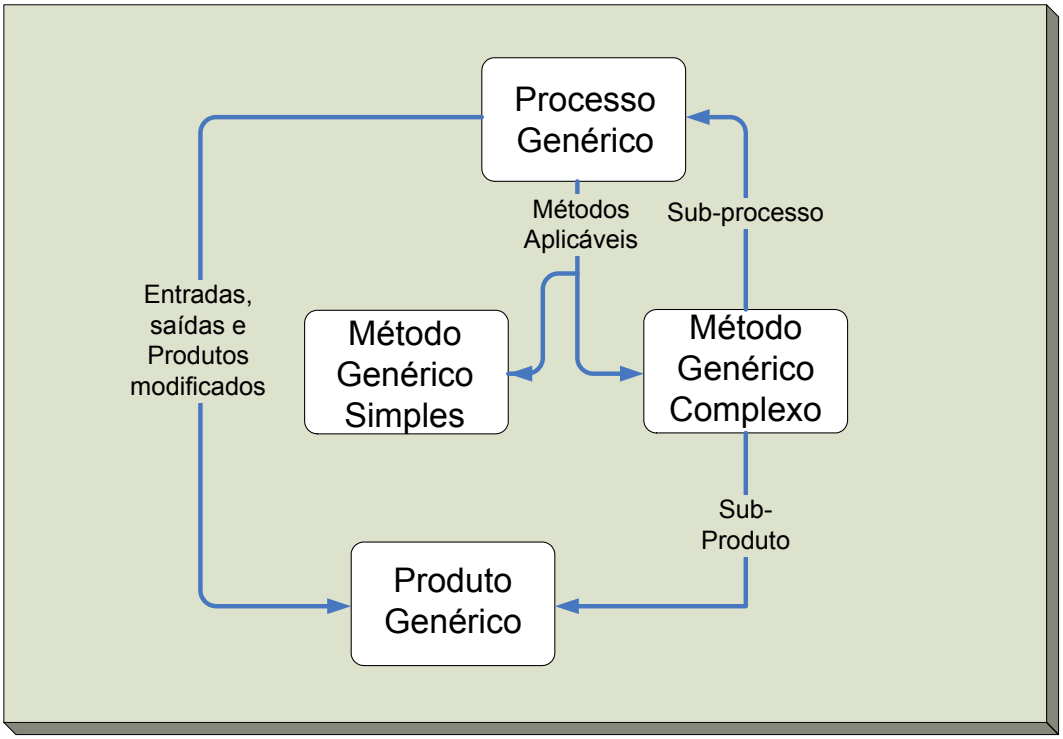


Figura 19 - Fichas para descrição de um Processo Genérico

Pode-se observar pela figura 19 que as fichas de um processo genérico têm referências às suas entradas, saídas e produtos do processo modificados. Cada produto é documentado através de uma ficha de descrição de produto genérico separada. Cada ficha de descrição de processo também contém *links* para um ou mais Métodos Genéricos. Este, por sua vez, pode ser Simples ou Complexo, observando que o método simples não possui qualquer referência à outra ficha. Processos, Produtos e Métodos são representados segundo a convenção da figura 20. Cada método Genérico Complexo conecta sub-processos a Processos Genéricos e sub-produtos a Produtos Genéricos. No próximo capítulo, maiores detalhes do processo de construção de um sistema CBR para Gerenciamento de Falhas em Redes de Computadores serão discutidos..

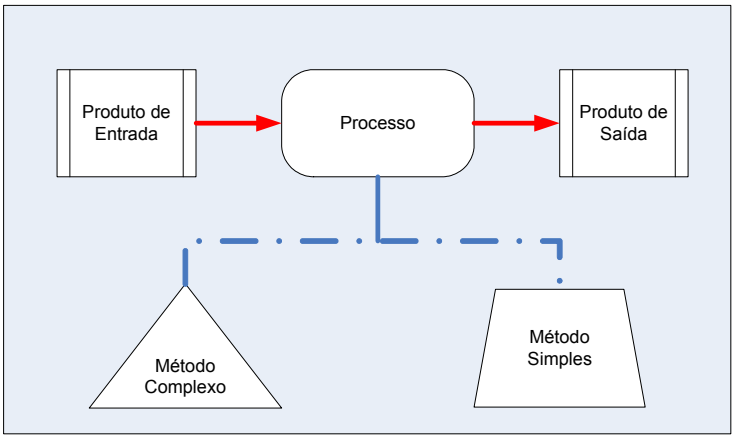


Figura 20 - Convenções da Metodologia INRECA II

5 Sistema Baseado em Casos para Diagnóstico de Redes

Neste capítulo, as considerações realizadas no capítulo 4, fundamentadas em aspectos teóricos e em experiência especialista, serão utilizadas para a criação da aplicação CBR, de acordo com a sua terminologia própria.

5.1 Terminologia:

Este universo em particular utiliza a terminologia ilustrada na figura 21 e descrita detalhadamente nas subseções que se seguem.

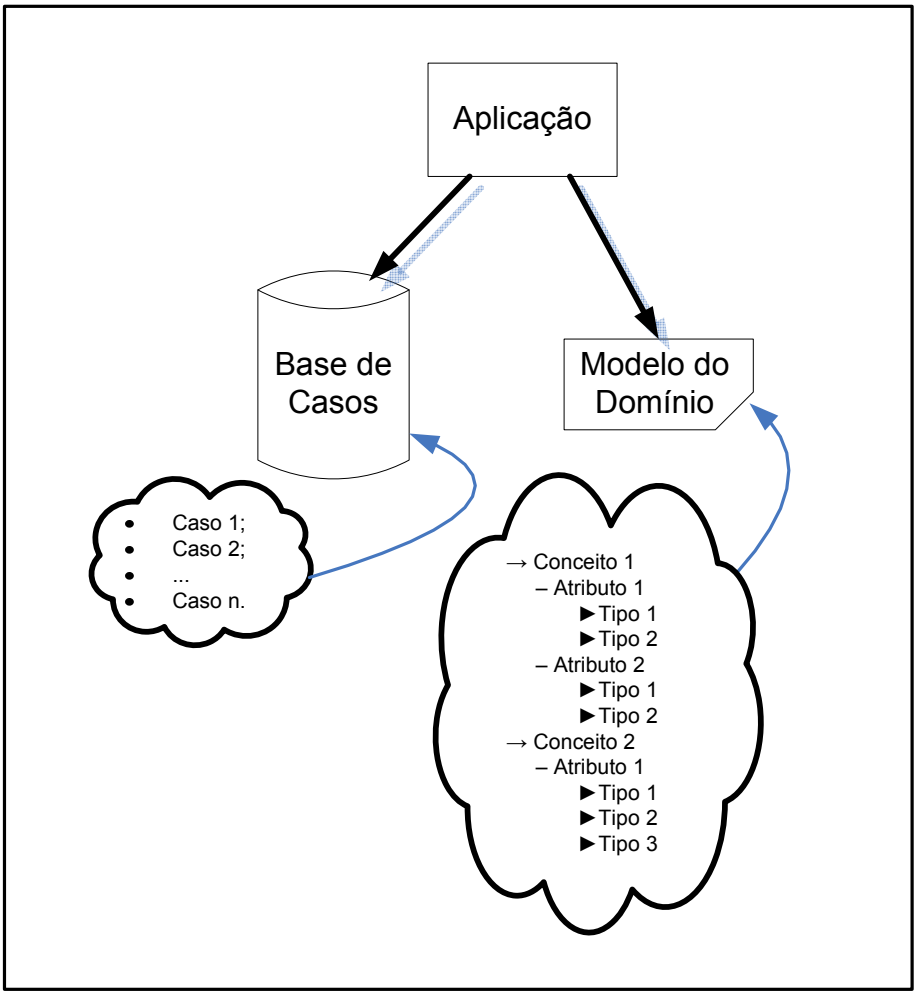


Figura 21 - Elementos Básicos da Modelagem de um Sistema CBR

A figura 21 permite a observação da estratégia para modelagem de sistemas CBR. Uma aplicação CBR é composta de uma Base de Casos, que são descritos através de um Modelo de Domínio. O "Domínio" é a área de discussão na qual os casos estão inseridos e para a qual a aplicação será desenvolvida. No caso deste trabalho, o Domínio é o Gerenciamento através do Diagnóstico de Falhas em Redes de Computadores. O

Modelo deste Domínio contém toda informação que permita a descrição das relações entre os dados dos casos e a sua terminologia, adicionando significado aos dados. Isto é fundamental na aplicação, já que este significado é necessário para a comparação e recuperação entre os casos disponíveis, a partir de um caso descrito através de seus atributos. Em uma abordagem simples e objetiva, pode-se simplificar a metodologia de construção de aplicações CBR através dos seguintes passos: inicialmente, o modelo do domínio em questão deve ser descrito através de seus conceitos, atributos e tipos. Em seguida, os casos são colecionados, preferencialmente em uma abordagem de protótipo, ou seja, com retorno para a primeira etapa sempre que alguma nova característica de um conceito for revelada e for relevante para a descrição dos casos. Isto é importante, já que durante a construção do modelo, eventualmente alguma característica pode não estar visível em uma primeira avaliação. As seções seguintes vão descrever o Modelo do Domínio da aplicação desejada através desta terminologia.

Algumas definições peculiares da terminologia usada no *CBR-Works* são necessárias para o entendimento da sua utilização e a criação de uma aplicação para diagnóstico de falhas dentro do seu ambiente de desenvolvimento:

- **Domínio** (*Domain*) – área de foco da aplicação em desenvolvimento. No caso deste trabalho, são Falhas em Redes;
- **Aplicação** (*Application*) – produto da implementação de características do domínio em questão. É constituída de uma Base de Casos (*Case Base*) e um Modelo de Domínio (*Domain Model*);
- **Base de Casos** (*Case Base*) – Banco de Dados onde os casos são armazenados. Uma descrição específica de falha é um caso;
- **Modelo de Domínio** (*Domain Model*) – Contém todas as informações que descrevem as relações entre os dados dos casos e a terminologia do domínio. Este modelo é a ferramenta para adicionar significado aos dados para que se possa haver uma comparação entre eles. A definição dos parâmetros de erro típicos em uma rede e das similaridades entre eles possibilita a comparação entre os diversos casos que possuam tais parâmetros. O Modelo é descrito através de conceitos (*concepts*), atributos (*attributes*) e Tipos (*Types*), definidos a seguir;

- **Conceitos** – Um evento específico, produto da busca feita pela aplicação em desenvolvimento. É um diagnóstico de falha;
- **Atributos** – São características, parâmetros, elementos que pertencem ao conceito em descrição e que o definem;
- **Tipos** – Estes atributos podem ter ranges específicos de possibilidades, definidos através do seu "tipo". Como a maioria dos sintomas característicos de falhas é binária, ou seja, estão presentes ou não, na aplicação de diagnóstico de falhas o tipo mais utilizado é o *boolean*. Após definido o Modelo do Domínio de falhas, passar-se-á para a fase da coleção de casos para alimentação da base.

A seguir, os elementos do foco da pesquisa deste trabalho serão adequados à terminologia acima descrita:

5.1.1 Domínio (*Domain*)

Gerenciamento de Falhas em Redes de Computadores

5.1.2 Modelo do Domínio (*Domain Model*)

Composto dos Seguintes Aspectos:

- **Conceito** – Diagnóstico
  - **Atributos:**
    - ✓ Descrição
    - ✓ Sintomas
    - ✓ Sinais
    - ✓ Testes confirmatórios
    - ✓ Tratamento
  - **Tipos** (*Types*) – São os *ranges* dos atributos:

- **Descrição** – trata-se de um campo textual, não usado para o diagnóstico, uma vez que apenas detalha o mesmo. Os sintomas, sinais e testes são dependentes das subáreas do conceito. Como as falhas possuem tipicidade de acordo com a sua camada de ocorrência, e é conveniente para a atividade de suporte esta segregação, serão implementados subconceitos de acordo com as camadas OSI.
- **Sintomas**:
  - ✓ **Falhas Físicas**:

- Falta de Conectividade, Conectividade Intermitente, Rede Lenta, indicação de taxa incompatível com as taxas máximas da placa e do equipamento de conectividade;

✓ **Falhas de Enlace**:
- Falta de conectividade, Conectividade Intermitente, Rede Lenta;

✓ **Falhas de Rede**:
- Falta de conectividade com alguns destinos, especialmente fora do escopo da rede local; falta total de conectividade; falta seletiva de conectividade.

✓ **Falhas de Aplicação:**
- Rede Lenta

- **Sinais**:

✓ **Falhas Físicas:**
- Ocorrência de taxa de erros diferente de zero; taxa de colisões acima de 10%; ocorrência de colisões tardias; equipamento inoperante; interface com estado administrativo "*down*"; taxa de CPU acima de 75%; taxa de utilização de memória acima de 75%; tráfego *broadcast* e/ou *multicast* elevado; alteração (aumento) inesperado da taxa média de ocupação do enlace, quadros na rede com tamanho superior a 1518 bytes, taxa de ocupação de barramento *half-duplex* superior a 50%; taxa de ocupação de barramento *full-duplex* superior a 70%;

✓ **Falhas de Enlace:**
- Taxas de Tx e Rx zeradas, apesar da interface estar aparentemente funcional; tráfego de *broadcasts* elevado; tráfego ARP elevado;

✓ **Falhas de Rede:**
- Existência de mensagens de redirecionamento na rede; existência de pacotes na rede (especialmente de *broadcasts*) com endereços de origem não-pertencentes ao escopo da rede local em análise; requisições ARP para endereços de destino não-pertencentes à rede local; o uso do endereço IP, ao invés do seu respectivo nome, em um *browser*, é bem sucedido, enquanto que com o nome não.

✓ **Falhas de Aplicação:**

- Existência de quadros ARP em profusão, emitidos por uma mesma estação de rede.

- **Testes:**
  - ✓ **Falhas físicas:**
    - *Leds* de equipamentos ou placas indicando falhas; equipamento de teste indicando descontinuidade do cabo; equipamento de conectividade ou cabos indicando alto nível de NEXT, atenuação ou ruído; *ping* no equipamento mais próximo não responde; *ping* no equipamento mais próximo dá tempo de resposta alto; *ping* no equipamento mais próximo dá erros; *ping* no equipamento mais próximo dá erros quando o cabo é movimentado; modo de operação (*full* ou *half-duplex*) incompatível entre dois dispositivos; verificação do estado administrativo de interfaces com software de gerência indica discrepância; a substituição do equipamento ou componente sob suspeita resolve o problema; verificação do equipamento sob suspeita com software de gerência indica alta taxa de ocupação de CPU; verificação do equipamento sob suspeita com software de gerência indica alta taxa de broadcast; verificação do equipamento sob suspeita com software de gerência indica alta taxa de *multicast*; verificação do equipamento sob suspeita com software de gerência indica alta taxa de uso de memória; verificação do *driver* do dispositivo de conectividade (se é o do fabricante e está atualizado), verificação da rota do cabo, com relação a EIA/TIA 568A e seus apensos (ANÔNIMO, 2001), verificação se o desligamento do equipamento de reclamante faz cessar ou reduzir sinais de anormalidades (como taxa de erros), verificação (identificação) do causador de colisões; identificação de máquinas (IPs) com geração excessiva de *broadcast* (acima de um *broadcast* a cada 10 segundos, em média); verificação da indicação da taxa praticada no enlace, através da cor de *leds* ou por software; verificação visual dos conectores (arranjo dos fios); certificação primária do cabeamento com scanner; verificação, através de softwares e de inspeção, se a topologia da rede está alterada à

revelia dos controles de conformidade com as normas de *cabling* (ANÔNIMO, 2001).

- ✓ **Falhas de Enlace:**
  - A aferição do estado administrativo de uma interface indica "*down*"; aferição da taxa de broadcast, tendo como parâmetro de normalidade até um broadcast a cada 10 segundos por estação de rede; taxa de utilização de CPU de estações e equipamentos de conectividade superior a 75%; taxa de utilização de enlaces superior à média tipicamente observada; verificação do tempo configurado nas tabelas MAC x porta de comutadores para a variável *dot1AgingTime* buscando valores fora do *default* (300 segundos); número de quadros ARP em um domínio de colisão superior ao número de estações ativas por segundo; verificação do tempo configurado para a validade da *cache* ARP em estações diferente de 600 segundos (default);
- ✓ **Falhas de Rede:**
  - Verificação da configuração do *default gateway* da estação; busca de mensagens de ICMP *redirect* através de um *sniffer*; verificação do endereço IP configurado na estação; verificação da máscara configurada na estação; verificação do endereço de DNS *server* configurado na estação.
- ✓ **Falhas de Aplicação:**
  - Verificação da existência de muitas conexões TCP abertas, sem a existência e servidores correspondentes ativos; execução "limpa" de um antivírus.

- **Soluções:**
  - ✓ **Falhas Físicas:**
    - Troca do cabo; reparo do cabo ótico; certificação da rede, com correção de discrepâncias; reconectorização; reconfiguração manual do modo de operação; troca do equipamento; atualização do IOS (Sistema Operacional) do equipamento de conectividade; verificação do sistema de alimentação, aterramento e refrigeração do equipamento; reinstalação de *drivers* para os dispositivos de conectividade sob suspeita; limpeza de contatos e recolocação de placa de rede em outro *slot*; troca da placa de rede; lançamento de

cabo por outra rota, livre de interferências; lançamento de cabo por outra rota, com metragem de cabo de acordo com os padrões para a categoria desejada; substituição de *hubs* por *switches*; implementação de VLANs; segmentação em subredes;

- ✓ **Falhas de Enlace:**
  - Alteração do estado administrativo para "*up*"; segmentação do domínio de *broadcasts*; redução das arquiteturas de protocolos; reconfiguração do tempo de envelhecimento das entradas MAC x porta em *switches*; reconfiguração do tempo de vida da *cache* ARP.
- ✓ **Falhas de Rede:**
  - Configurar corretamente o *default gateway* da estação; configurar corretamente o IP da estação; configurar corretamente a máscara de subrede da estação sob suspeita; configurar corretamente o endereço do DNS *server*.
- ✓ **Falhas de Aplicação:**
  - Limpeza da estação contaminada; atualização das vulnerabilidades do Sistema Operacional; instalação de um antivírus atualizado.

- **Diagnósticos** – Cabo interrompido; conectores irregulares; Incompatibilidade de modo/taxa de operação entre Interfaces; equipamento de rede defeituoso; placa de rede defeituosa; cabo sob interferência eletro-magnética; congestionamento do barramento; Uso de cabo incorreto; descumprimento de regras Ethernet. Interface desabilitada; tempestade de *broadcast*; tempo de envelhecimento de entrada MAC x porta de *switch* inadequado; tempo de envelhecimento de entradas ARP em estações inadequado; Configuração incorreta do *default gateway*; tabela de *hosts* de roteadores incorreta; endereço IP incorreto; máscara de subrede incorreta; endereço de DNS *server* incorreto ou indisponível; contaminação por vírus.

Para inserir os casos na ferramenta *CBR-Works*, criando uma aplicação para auxílio no diagnóstico de falhas em redes, é necessária a precisa definição do modelo do

domínio, que por sua vez definirá a estrutura do banco de casos. Este modelo contém todas as informações descrevendo as relações entre os casos e a terminologia específica do domínio em questão, adicionando assim um significado aos dados que será necessário para comparação entre os demais. A similaridade entre eles é um significado comparativo importante, por permitir a seleção dos casos com maior chance de acerto dentre os disponíveis. Conforme já descrito no início deste capítulo, o modelo é descrito através de conceitos (neste domínio, o diagnóstico da falha), atributos (sintomas e sinais) e tipos, que poderá ser apenas um *flag* de presente ou não (rede lenta) ou um *range* específico (percentual de *broadcasts*, de 0 a 100% do tráfego). Para particularizar ainda mais a aplicação, usando um recurso disponível na ferramenta, serão implementados os subconceitos referentes às camadas TCP/IP onde as falhas mais ocorrem, ou seja, física, enlace, rede e aplicação. Cabe ressaltar que a ferramenta utilizada para implementação do modelo restringe os tipos a: *boolean, date, integer, real, set, string, symbol, time e timestamp*. Tipos apropriados devem ser criados como subtipos de algum dos tipos disponíveis, implementando assim uma restrição ao range original. Esta restrição pode ser feita através da definição de um intervalo ou da enumeração dos tipos possíveis.

Ao se definir o conceito e seus respectivos subconceitos, passa-se para a fase da definição, para cada subconceito, das propriedades de cada um. Tais propriedades são apenas descritivas, sem função na seleção de casos. Os atributos de cada subconceito descreverão então os sintomas e sinais presentes nas falhas típicas, obtidos através da experiência especialista do autor, enriquecida por documentação técnica da área e os casos reais coletados no *test-bed* utilizado.

Além dos atributos indicativos de determinadas falhas, ainda podem ser definidas métricas de similaridade e regras para adaptação de casos, cujas considerações sobre a utilização serão apresentadas mais adiante. As questões a serem apresentadas ao usuário da aplicação são elaboradas através da última aba do subconceito em construção. Para que os atributos de cada subconceito possam ser introduzidos, é necessária a definição prévia dos tipos e subtipos adequados, de acordo com o descrito no parágrafo anterior. Foram então definidos os subtipos descritos na tabela 3.

Tabela 3 - Tipos de detalhamento de cada atributo

Descrição	Sub-tipo	Tipo	Range
Estado da Rede	Estado	Symbol	fora, lenta, intermitente
Taxa de erros	Tx Erro	Boolean	sim, não
Taxa de colisões acima de 10%	Tx Colisão	Boolean	sim, não
Taxa de CPU acima 75%	Tx CPU	Boolean	sim, não
Tráfego *broadcast* acima do normal[2]	Alto Broadcast	Boolean	sim, não
Quadros acima de 1518 bytes	Jumboquad	Boolean	sim, não
Taxa de ocupação do barramento acima do normal[3]	Tx Ocupação	Boolean	sim, não
Taxa de transmissão nula ou intermitente	Taxa Tx	Boolean	sim, não
Taxa de recepção nula ou intermitente	Taxa Rx	Boolean	sim, não
Tráfego ARP coletivo intenso[4]	Alto ARP geral	Boolean	sim, não
ICMP Redirect	ICMP Redirect	Boolean	sim, não
Pacotes com endereços de destino não-pertencentes à rede	Outpack	Boolean	sim, não
Tráfego ARP de uma estação intenso[5]	UniArp	Boolean	sim, não
Muitas Conexões TCP abertas[6]	TCPopen	Boolean	sim, não
Diagnóstico da Falha	Diagnostico	String	texto livre

Apenas houve a necessidade de se criar o subtipo *Estado*, que limita os símbolos às possibilidades enumeradas, ou seja, *fora*, *lenta* ou *intermitente*. Os demais foram criados usando o tipo *boolean*. Na criação do subtipo estado, utilizou-se um recurso importante na recuperação de casos, o editor de similaridade. Para o usuário, às vezes é imprecisa a noção do estado da rede, especialmente entre rede "lenta" ou "intermitente". Já entre "fora" ou "intermitente" e "lenta", já é mais fácil a percepção. Por conta disso, optou-se por particularizar o grau de similaridades entre estes, de acordo com a experiência especialista do autor, usando o recurso "*table*" do modo de similaridade entre atributos *standard*. Esta atribuição está ilustrada na figura 22.

[2] Como já citado neste trabalho, considera-se razoável a emissão de 1 *broadcast* a cada 10 segundos por cada máquina no barramento compartilhado.
[3] 50% em barramento *half-duplex* e 70% em barramentos *full-duplex*.
[4] Quadros ARP em número superior ao número de estações por segundo no barramento compartilhado.
[5] O uso de um *sniffer* indica *top talkers* com broadcasts ARP atípicos, superiores ao dobro de servidores no mesmo barramento.
[6] A existência de conexões abertas sem demanda justificada, como a execução de servidores é sintoma de programas maliciosos

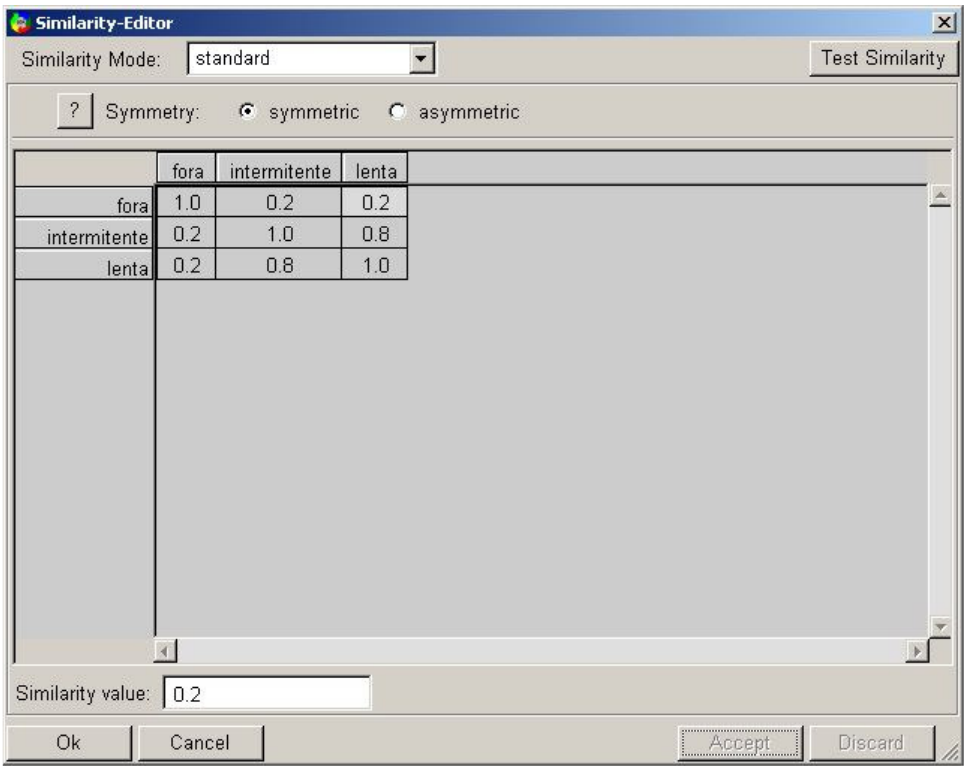


Figura 22 - Tabela de similaridade entre atributos

Após a criação dos tipos no *type editor* do CBR-Works, passa-se para a fase da caracterização dos atributos do modelo em discussão, no *Concept Manager*. Para a inserção dos atributos, foram necessárias decisões a respeito de dois aspectos: *discriminant* e *mandatory*. Um atributo é marcado como *discriminant* quando ele é parâmetro que permite a distinção de um caso dos demais. Se o atributo não for discriminante, será ignorado no cálculo da similaridade. O atributo seguinte é o *mandatory* , marcado quando a sua utilização é obrigatória para todos os casos. Seria o caso da eleição dos atributos mais relevantes como mandatórios, atribuindo-se, além disso, um outro aspecto do atributo, o seu peso (*weigth*), que define o grau de importância que um atributo possui em relação aos demais, no momento da recuperação de casos similares. Um exemplo da utilização deste peso diz respeito aos problemas de aplicação relacionados com infecções virais, cuja indisponibilidade de antivírus possui grande peso na indicação das causas de lentidão por contaminação, conforme demonstrado no estudo de caso no capítulo dos resultados deste trabalho. Por tratar-se de uma tela capturada ainda no início do trabalho, os pesos foram mantidos iguais, para ajuste durante a fase de refinamento, de acordo com a metodologia INRECA (capítulo 5). A figura 23 ilustra a inserção dos atributos e suas características citadas.

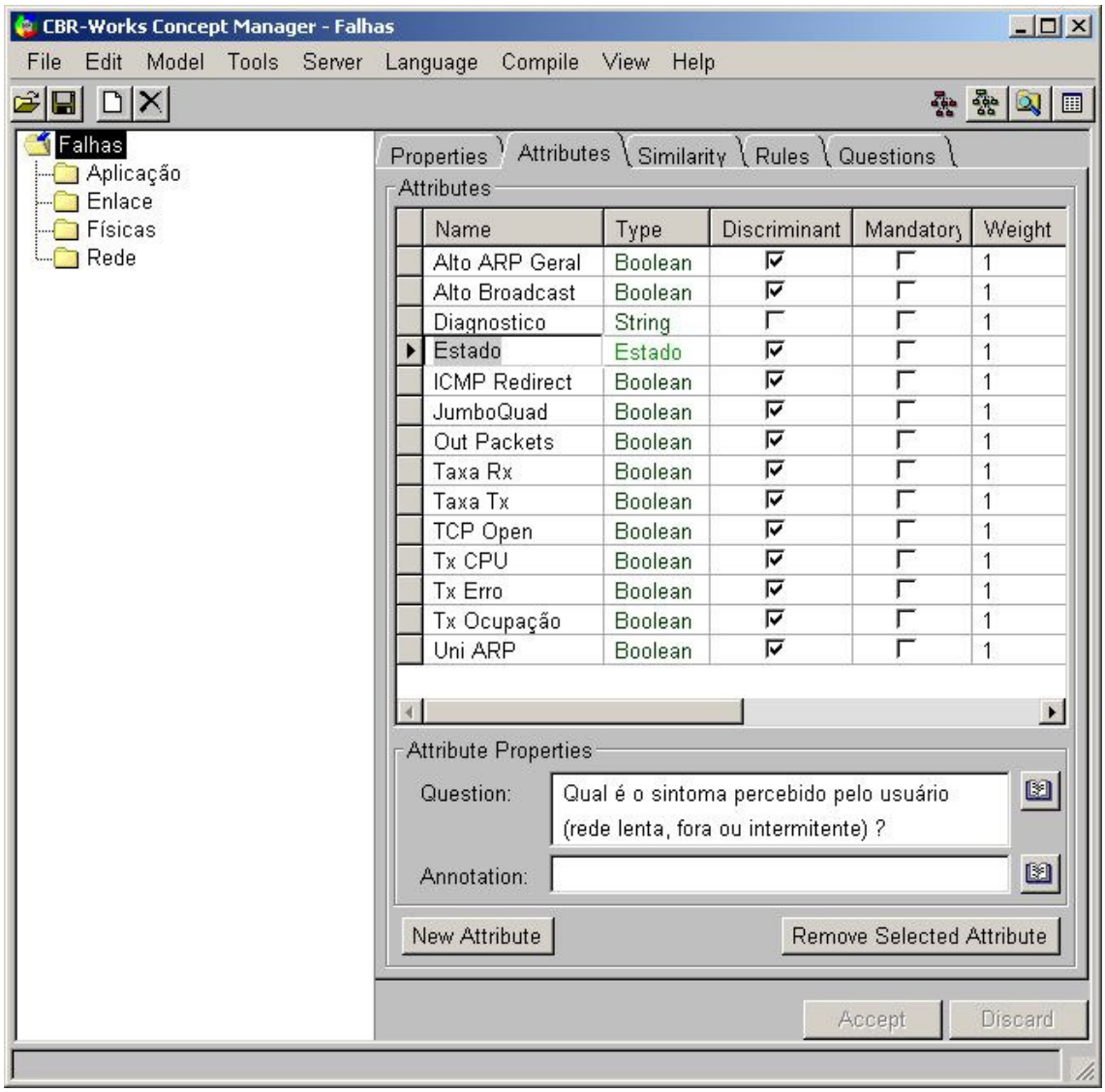


Figura 23 - Atributos dos subconceitos

A similaridade é outro aspecto de relevância fundamental para o sucesso de um sistema CBR. O *CBR-Works* implementa duas componentes para o cálculo da similaridade entre os conceitos (casos). A similaridade entre o conteúdo dos conceitos e a similaridade entre a estrutura dos conceitos. A similaridade baseada no conteúdo é computada com base nos seus atributos, podendo ser utilizada uma das seguintes métricas:

- *Average* – Quando todos os atributos contribuem para o cálculo da similaridade global através da média aritmética;
- *Euclidean* – Uma interpretação geométrica no cálculo da similaridade, através do computo da distância entre dois conceitos baseada nos seus respectivos conceitos;
- *Minimum* – A menor similaridade entre os atributos dos conceitos define a similaridade final; e

- *Maximum* – A maior similaridade entre os atributos dos conceitos define a similaridade final.
- *Custom* – a similaridade adequada à aplicação em elaboração pode ser definida pelo usuário através de regras ou formulação própria programada de forma semelhante ao Pascal.

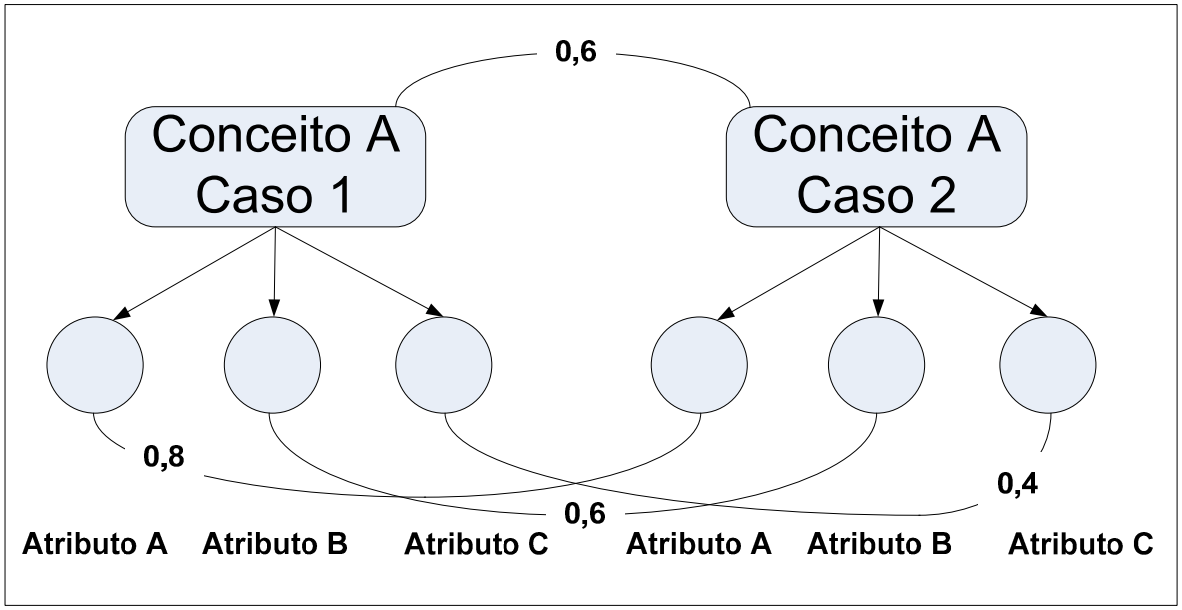


Figura 24 - Similaridade média entre os atributos de um mesmo conceito

Para o caso da similaridade *Average*, são computadas as similaridades entre atributos correspondentes de dois casos. A média aritmética das similaridades apuradas é então a similaridade entre o conteúdo dos casos. A figura 24 ilustra a similaridade entre dois casos que possuem três atributos idênticos e que a similaridade entre estes atributos é variável. No método *Euclidean*, é usada a média geométrica, e os métodos *maximum* e *minimum* são auto-explicativos. Estes quatro métodos disponíveis na ferramenta CBR-Works permitem a implementação da similaridade com base no conteúdo dos conceitos (casos). Já a similaridade entre a estrutura dos conceitos permite o cômputo da similaridade independentemente da correspondência entre os atributos. Para isso, a similaridade entre conceitos dissimilares pode ser definida implícita ou explicitamente, com base em uma visão taxonômica da hierarquia entre os mesmos, conforme ilustração da figura 25.

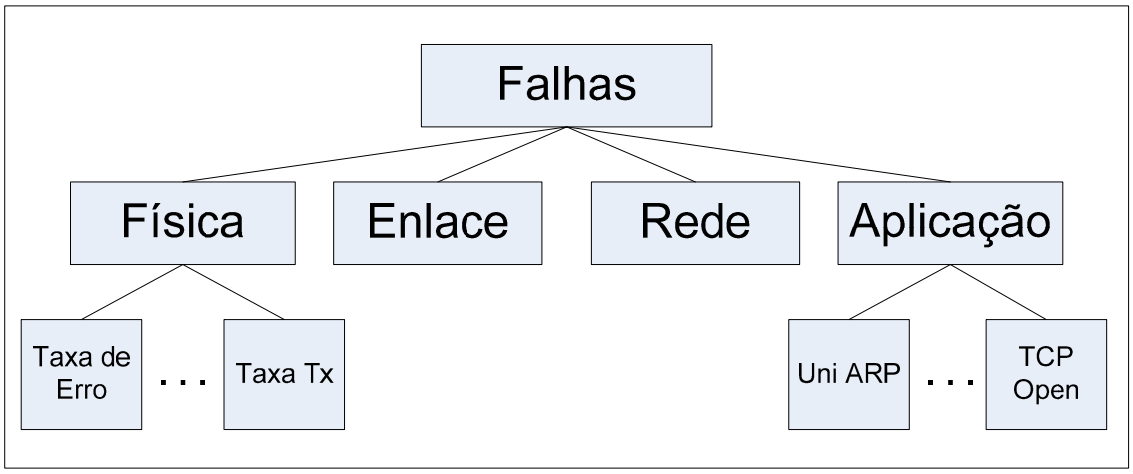


Figura 25 - Visão Taxonômica dos Atributos

O cálculo da similaridade baseado na estrutura pode ser feito através da definição, inicialmente, do modo de similaridade, entre as opções:

- *Standard* – após a definição da simetria, detalhada no próximo parágrafo, parte-se para a definição do significado da estrutura taxonômica adotada. Caso nodos internos às folhas possuam significado passível de mensuração e busca, é necessária a definição da semântica dos nós internos, o que não é o caso deste trabalho;
- *Advanced* – agrega a capacidade de definir a similaridade genérica entre os clusters de símbolos na hierarquia taxonômica; e
- *Table* – possibilita um detalhamento maior das similaridades para cada par de atributos, inclusive de forma assimétrica. É o mais adequado para a situação deste trabalho.

A assimetria citada acima é usada quando existe diferença na importância de um determinado atributo presente na *query* e o mesmo atributo presente no caso. Essa facilidade é útil, por exemplo, caso se afira num determinado evento, a existência de Alto *Broadcast*, e que se considere a similaridade com casos da base que possuam, por exemplo, ARP intenso, maior do que o inverso, ou seja, *queries* com ARP intenso ter similaridades inferiores com casos onde haja Alto Broadcast, em função da eventual maior facilidade em se detectar o Alto *Broadcast* que o ARP. Essa eventual assimetria pode ser caracterizada na aplicação, conforme visto na figura 22.

Na edição das preferências para a aplicação em desenvolvimento, é possível assinalar os percentuais de cada componente no cálculo da similaridade final entre os casos. O *Default* é 100% da similaridade entre os atributos e 0% estrutural. A figura 26

demonstra o resultado da transposição de alguns subconceitos com seus respectivos atributos para o *CBR-Works*.

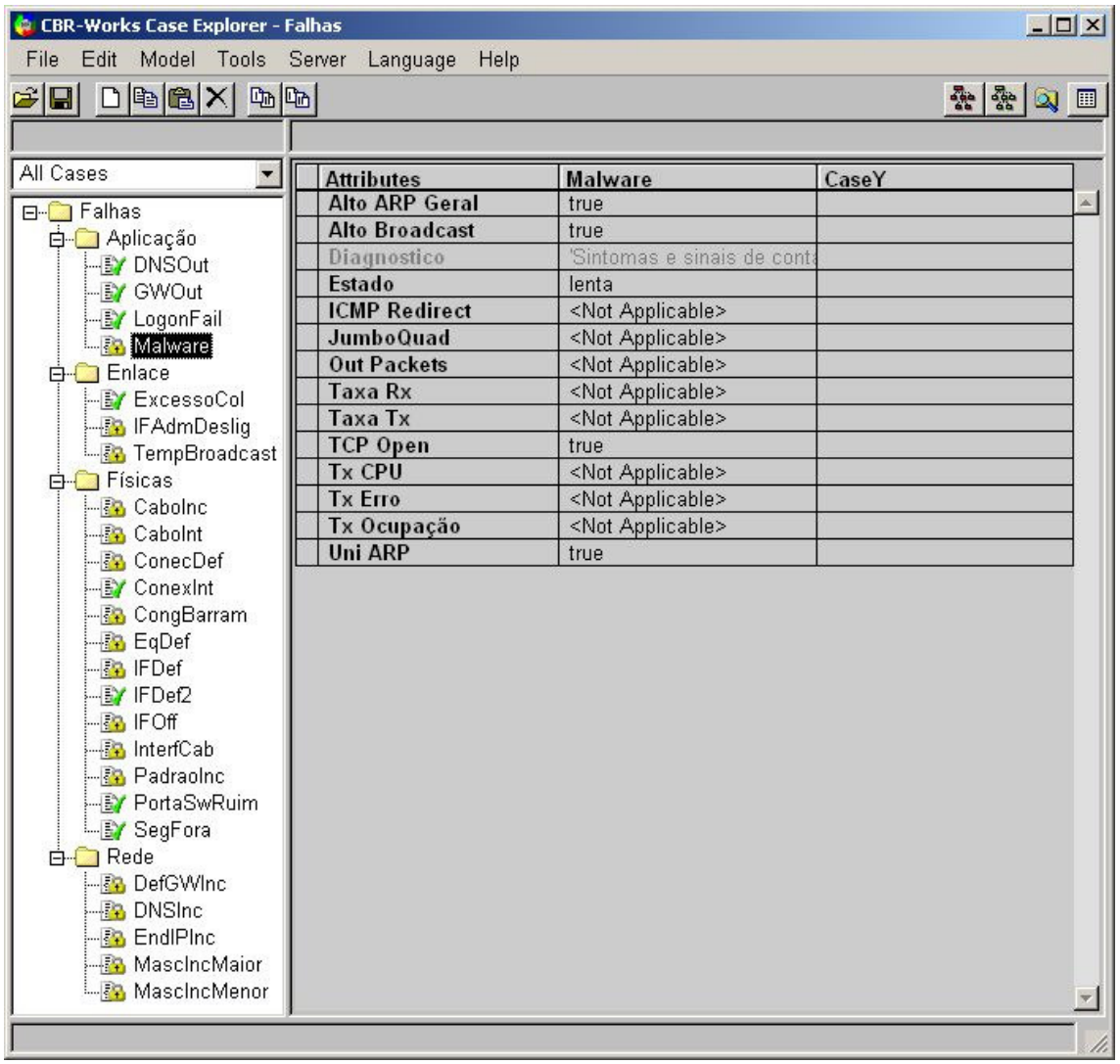


Figura 26 – Alguns sub-conceitos Implementados

Após a definição do modelo do domínio das Falhas, são inseridos os resultados da experiência especialista e da pesquisa sobre elementos (parâmetros) obteníveis da rede que são típicos das falhas mais importantes. Isto é feito através do *Case Explorer*, como casos protegidos. Estes tipos de casos, além de serem usados na recuperação de casos similares, são protegidos de alterações indevidas. Além deste tipo, há também os casos não-confirmados, quando os mesmos estão incompletos (nem todos os parâmetros da falha puderam ser aferidos) ou ainda não foram confirmados em seu diagnóstico. Apesar de disponíveis na base, não são considerados no cálculo das similaridades para recuperação. Novos casos, completos e confirmados, são categorizados como "*confirmed*", ficando disponíveis para recuperação. Casos obsoletos podem ser

mantidos para fins de estatísticas sem prejudicar a objetividade na recuperação dos casos válidos, bastando para isso serem classificados como "*obsolete*".

Os casos primitivos, ou seja, aqueles oriundos da experiência especialista, foram inseridos com nomes bem sugestivos, sem índices. Além disso, foram categorizados como *protected*, já que estão livres de interpretações equivocadas. Após este trabalho preliminar, foram inseridos casos reais acumulados durante aproximadamente seis meses de registro de ocorrência de problemas reais no *test-bed* utilizado, que não haviam passado pela pesquisa no CBR-Tools (fase de desenvolvimento). O resultado final desta fase foi a criação de uma base com 17 casos protegidos e 32 confirmados para os testes em campo.

Para uma adequada definição dos requisitos da aplicação CBR, que possibilitasse a sua documentação de forma padronizada, disponível para todos os membros da equipe, foi utilizada a metodologia INRECA, na sua versão II. O uso desta metodologia possibilitou o acompanhamento das metas, o cumprimento de todos os passos em cada processo por todos os envolvidos e, principalmente, a verificação de conformidade do planejado com o executado. A próxima seção documenta estas tarefas.

5.2 Adaptação da Metodologia INRECA

Para a modelagem do sistema CBR com o uso da metodologia INRECA, são necessários os seguintes passos (BERGMANN, ALTHOFF, 1998):

1. Definição da estrutura dos casos
2. Coleção de casos
3. Definição da importância de cada atributo
4. Definição de relacionamentos
5. Definição de similaridades
6. Construção da interface homem-máquina
7. Integração da aplicação aos sistemas legados
8. Refinamento da aplicação CBR

Esses passos conduzem a quatro principais tarefas, ilustradas na figura 27.

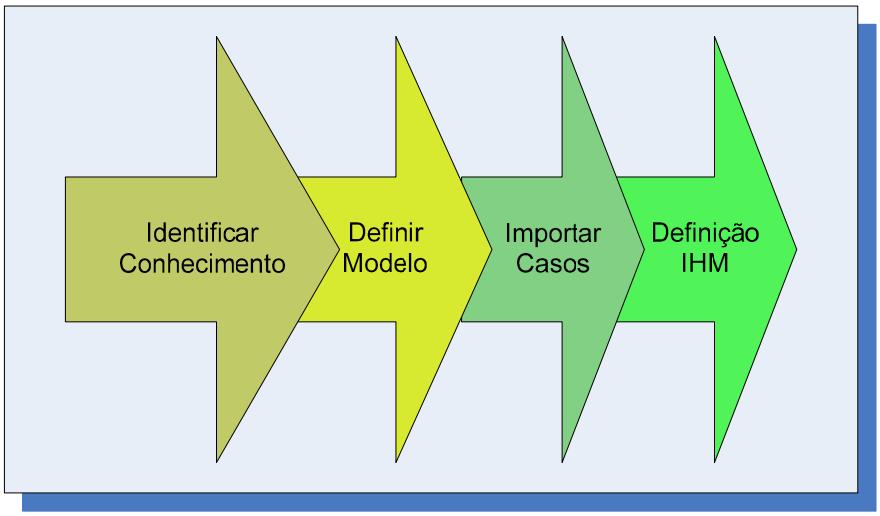


Figura 27 - Tarefas para desenvolver um CBR

O conhecimento a ser usado será o do universo de Gerenciamento de Redes, baseado em documentação especializada e, principalmente, em conhecimento especialista, descrito no capítulo 2.

A definição da estrutura do caso será obtida através do processo definido no capítulo 4, de acordo com a metodologia INRECA II. Através deste processo será definido o modelo do domínio, com os seus tipos, objetos, componentes, especializações, similaridades e regras. A importação dos casos será feita com o uso de um *test-bed* real, uma rede de uma Instituição de Pesquisas com uma rede de computadores com aproximadamente 400 (quatrocentas) máquinas, e uma diversidade de problemas e situações de falhas que demandam ações rápidas e pró-ativas no sentido de diagnosticar e contingenciar a falha, com o objetivo de evitar a propagação do erro e a perda de funcionalidades na rede. A IHM (Interface Homem-Máquina) será construída a partir das facilidades disponíveis no aplicativo *CBR-Works*.

O Nível Genérico Comum cobre métodos, processos e produtos relacionados a aspectos gerenciais, técnicos e organizacionais do desenvolvimento de uma aplicação CBR. Como neste trabalho será usada uma ferramenta disponível, será iniciado o esforço a partir do nível intermediário das figuras 17 e 18, o "livro de receitas". Neste nível, processos, produtos e métodos são focados no gerenciamento de falhas em Redes de Computadores. Os processos serão conectados para obtenção de um fluxo de produto que resulte em um diagnóstico e uma possível solução para o problema. Esta metodologia já foi testada em situações semelhantes (GÖKER *et al.,* 1998), visando a criação de um repositório de conhecimento para atendimento *Help-desk* de um domínio complexo.

Para a modelagem do Sistema de Suporte ao Gerenciamento de Falhas em Redes, será utilizada uma solução baseada em uma "receita" do projeto INRECA, direcionada para o desenvolvimento de sistemas CBR para *Help-Desks* (BERGMANN, GÖKER, 2003). Essa receita divide um sistema semelhante em quatro sub-processos;

- Planejamento e inicialização – Definições de objetivos, domínio inicial, e definição de requisitos para o sistema;
- Protótipo e avaliação – Teste da validade dos objetivos definidos na fase anterior;
- Implementação – Aquisição dos casos, ajustes finais;
- Utilização – disponibilizar para uso.

Estes sub-processos e seus inter-relacionamentos podem ser visualizados na figura 28.

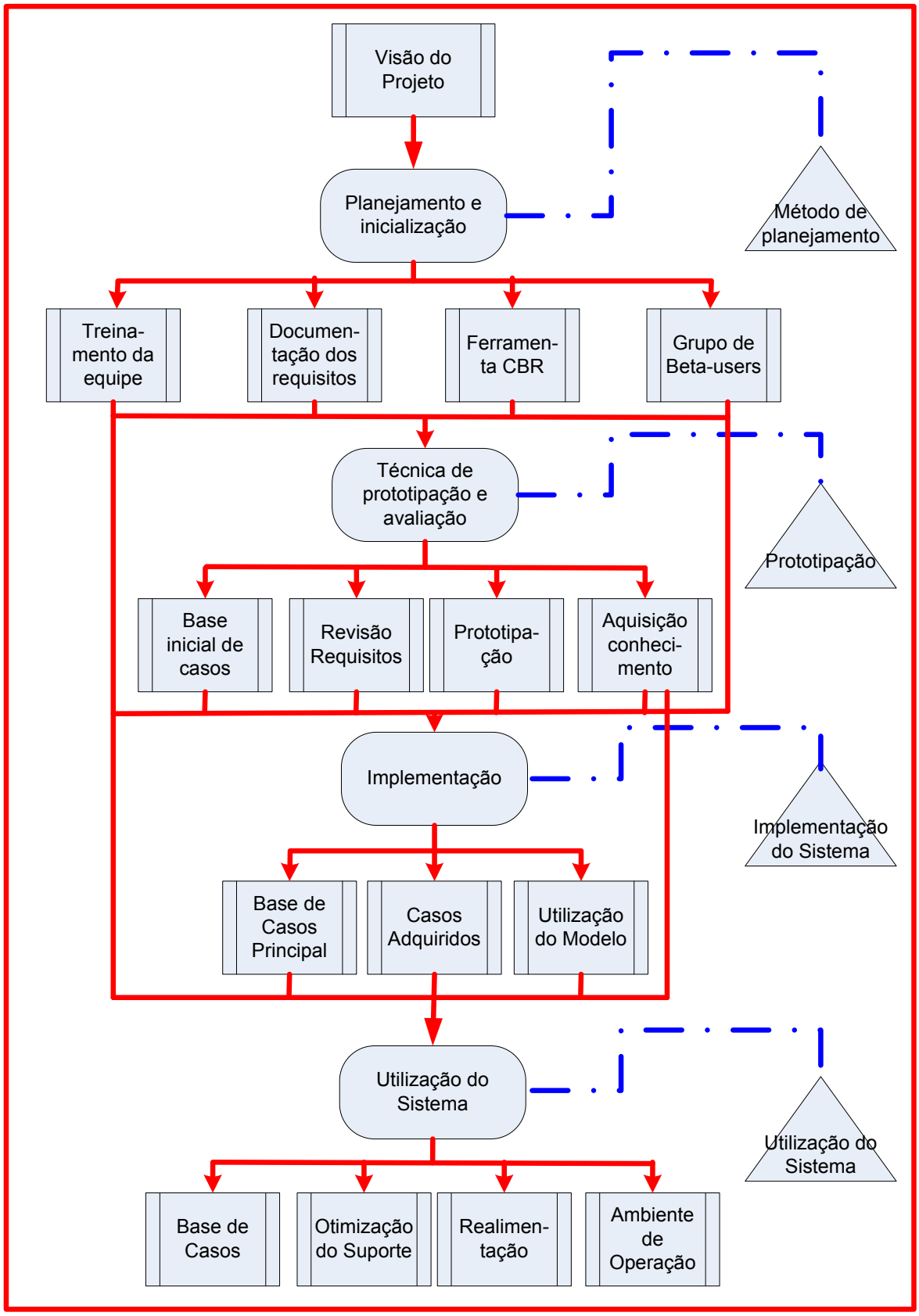


Figura 28 - Modelagem de Processo aplicada ao Gerenciamento de Redes

Na figura 28 são descritas, na forma de *workflow*, todas as etapas do planejamento, implementação e utilização do sistema. As definições apresentadas no capítulo 4 são

utilizadas aqui. Em síntese, os processos, identificados pelos retângulos com vértices curvilíneos, representam as atividades a serem executadas para transformar um produto de entrada em um outro de saída. Observando a figura 28, pode-se ilustrar o processo "Técnica de Prototipação e Avaliação", que realiza com os insumos "Equipe treinada", "Requisitos", "Ferramenta CBR" e "*Beta-users*", alguns passos pré-definidos para obter os produtos "Base inicial de Casos", "Aquisição de Conhecimento", "revisão de Requisitos" e "Prototipação". Os processos têm seus respectivos objetivos, métodos para alcançar os objetivos e recursos, que podem ser agentes (humanos ou equipes de humanos designados para realização das tarefas de um processo) ou ferramentas. Um exemplo de ferramenta descrita nesta fase foi a aplicação para descoberta de informações da rede que será apresentada na subseção 5.3.2. Os produtos, identificados por retângulos com barras paralelas às arestas verticais, identificam os resultados produzidos pelos processos ou então elementos requeridos pelos processos para produção de novos produtos. Pode-se então classificá-los em produtos de entrada ou de saída. Os métodos podem ser *simples*, identificados através de trapézios, ou então *complexos*, representados por triângulos. Os métodos simples apenas descrevem de forma objetiva o que fazer para alcance dos objetivos de um processo, os métodos complexos devido à necessidade de um maior nível de detalhamento, é composto de sub-processos. Toda esta documentação é gerada e disponibilizada em formato de página *web*, visando facilitar o acesso e a navegação. A figura 28 seria a página principal (index.html), através da qual as demais são acessadas. Clicando na figura "Planejamento e Inicialização do Projeto", um *hyperlink* aciona o seu conteúdo, ilustrado na figura 29.

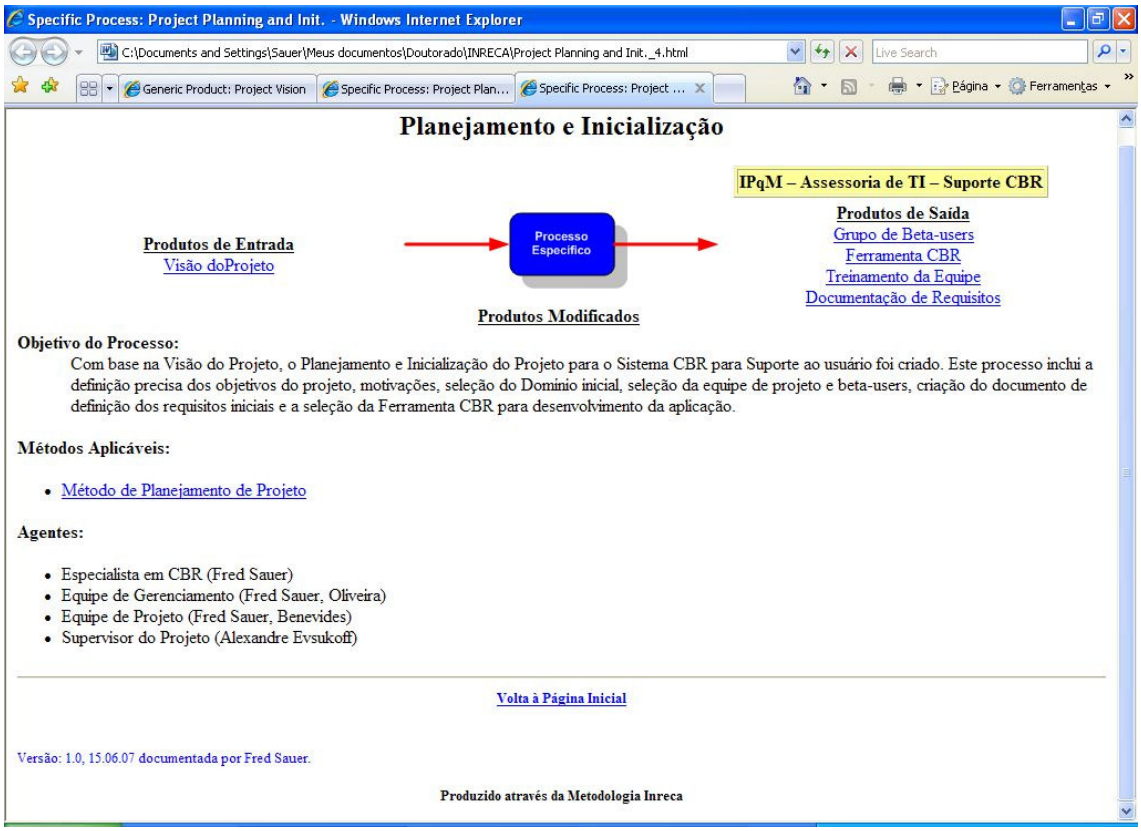


Figura 29 - Ficha da Documentação INRECA

Na figura 29, pode-se perceber que há outros *hyperlinks* para navegação, de forma que a documentação é dinâmica e fácil para acompanhamento e entendimento do sistema desenvolvido. Algumas outras páginas relevantes da documentação do sistema CBR desenvolvido neste trabalho estão disponíveis no apêndice 2.

A utilização da metodologia INRECA trouxe vários benefícios. Primeiro, por tratar-se de um sistema específico cuja "receita" já estava disponível, manteve compatibilidade com outros sistemas já testados em outras instalações, como o caso do Sistema HOMER da *Daimler* alemã (BERGMANN, GÖKER, 2003). A partir daí, de acordo com a metodologia, buscou-se a particularização em função dos objetivos, ambiente (domínio e modelo de domínio) a atender, e outras características particulares, para construir o nível específico de projeto. Este detalhamento, documentado em formato de hipertexto em páginas *html*, facilitou a busca da conformidade do sistema com os requisitos estabelecidos, bem como acelerou o processo de desenvolvimento e validação do sistema, por estarem os passos necessários previamente definidos e detalhados. Outro aspecto relevante é que, com o uso desta metodologia, mantém-se compatibilidade com um forte segmento acadêmico em atuação na Europa,

possibilitando a agregação de valor ao estudo dos sistemas de Raciocínio Baseados em Casos.

Este capítulo apresentou detalhes acerca da técnica para documentação de sistemas CBR mais discutida na literatura, com várias aplicações já documentadas e disponíveis. Visando manter compatibilidade com este esforço de padronização na documentação, a mesma foi adotada neste trabalho e alguns resultados estão ilustrados no Apêndice. A próxima seção discutirá as estratégias utilizadas neste trabalho para otimizar os objetivos principais da aplicação CBR, ligados à atividade de suporte aos usuários da Rede Local do Instituto de Pesquisas da Marinha.

5.3 Aplicações para Redução do Tempo de Indisponibilidade

Conforme observado pelo autor na sua experiência diária, o perfil típico do usuário das redes é o seguinte:

- Pouco conhecedor dos detalhes da infra-estrutura de conectividade;
- Impaciente na observação das falhas e suas possíveis razões, acionando o suporte sem tentar coisas simples, como verificar conexões físicas eventualmente desconectadas;
- Pouca percepção dos sintomas, confundindo facilmente "rede lenta" com "rede fora", conduzindo a perda de tempo no diagnóstico e solução de problemas;
- Desinteresse na observação de mensagens eventualmente surgidas na ocorrência da falha, que poderiam conduzir a um diagnóstico mais rápido; e
- Desconhecimento de ferramentas de simples manejo (*ping, tracert* e *ipconfig* no Microsoft Windows) que permitiriam um diagnóstico imediato em casos, por exemplo, de endereços configurados incorretamente.

Um questionamento imediato é: já que as redes estão crescendo exponencialmente, o nível de dependência do usuário destas redes é cada vez maior, e as equipes de suporte são pequenas e sempre sobrecarregadas, porque os Sistemas Operacionais não oferecem funcionalidades de suporte ao usuário para tarefas simples, como ler a configuração de rede, ou interpretar as mensagens recebidas na camada de rede, pelo protocolo ICMP (POSTEL, 2006), e alertar o usuário.

Para testar a viabilidade destas funcionalidades, foram desenvolvidas duas ferramentas de suporte ao usuário, passíveis de serem disponibilizadas dentro do ambiente operacional. Uma para captura dos parâmetros de configuração da rede e outra para tratamento do ICMP.

5.3.1 ICMP - Um Protocolo para Relato de Erros

O ICMP (*Internet Control Message Protocol*) é um protocolo *default* da camada de rede do TCP/IP. Isto significa que, instalando-se uma placa de rede ou um modem no computador, o ICMP estará presente. Este protocolo se presta a relatar erros ocorridos no processamento de datagramas[7]. As mensagens ICMP são despachadas em várias situações, como por exemplo, quando um datagrama não encontra o seu destino, quando um roteador não está com capacidade de transmitir o seu datagrama, por indisponibilidade de memória, ou quando o roteador recebe um datagrama para despacho externo, mas o destino se encontra no escopo da rede local. O Protocolo IP não foi criado para ser absolutamente confiável, e sim para realizar o seu melhor esforço no despacho dos datagramas. O propósito das mensagens de controle do ICMP é prover um relato de problemas no ambiente de comunicação, e não tornar o IP confiável. Não há garantias de que um datagrama será entregue ou uma mensagem de controle ICMP será recebida. Para evitar um "efeito cascata[8]" de mensagens e comprometimento da rede, não são enviadas mensagens ICMP sobre perda de mensagens ICMP. Algumas mensagens muito úteis do ICMP destinam-se a:

- **Identificar erros da rede** – Quando, por exemplo, torna-se impossível alcançar um *host* ou uma determinada sub-rede, devido a algum tipo de falha. Um pacote TCP ou UDP direcionado a um número de porta sem um destinatário determinado é também reportado via ICMP;
- **Anunciar o congestionamento da rede** – Quando um roteador começa a armazenar pacotes demais, devido a uma impossibilidade de transmiti-los tão rapidamente como estão sendo recebidos, gera-se uma mensagem ICMP de *Source Quench*. Direcionadas ao remetente, estas mensagens devem fazer com que a taxa da transmissão do pacote seja diminuída. Naturalmente, a geração de mensagens de *Source Quench* indiscriminadamente poderia causar ainda mais congestionamento na rede, assim, essas mensagens são usadas de forma limitada;
- **Ajudar na resolução de problemas** – O ICMP suporta uma função de *echo*, que envia apenas um pacote em uma viagem de ida e volta entre dois *hosts*.

[7] Datagrama – Unidade de dados da camada OSI de Rede, nível onde ocorrem as comunicações entre roteadores.
[8] O "Efeito Cascata" ocorre quando um evento causa outro não relacionado à falha original, dificultando o gerenciamento de falhas.

O *ping*, uma ferramenta comum de gerência de rede, é baseado nesta característica. O *ping* transmite uma série de pacotes, mensurando os tempos médios dos trajetos circulares e computando os percentuais de perda;

- **Anunciar *Timeouts*** – O campo TTL (*Time To Live*) corresponde ao número de *hops* (roteadores) total que uma informação pode percorrer. Ele é decrementado a cada *hop,* e quando chega a zero, o roteador descarta o datagrama e envia uma mensagem à fonte informando que a informação não chegou ao seu destino, utilizando o ICMP. O *TraceRoute* (comando *tracert* no Windows) é uma ferramenta que mapeia rotas da rede ao enviar pacotes com pequenos valores de TTL e verificando os anúncios de perda ICMP no tempo.

Uma mensagem de redirecionamento ICMP é utilizada quando o roteador determina que existe um caminho melhor para o pacote que acabou de ser enviado. Neste caso a implementação do protocolo de roteamento pode definir um novo caminho que melhor destine o pacote, mas, para que isto surtisse efeito, o Sistema Operacional deveria ser capaz de se reconfigurar para reparo da rota incorreta, o que não ocorre. desta forma, as máquinas não consideram este tipo de mensagem e continuam enviando dados pelo pior caminho.

Ao contrário dos protocolos da camada TCP (*Transmission Control Protocol*) e UDP (*User Datagram Protocol*) do transporte que operam sobre o IP, o ICMP é logicamente posicionado na mesma camada que o IP. A habilidade de compreender o ICMP é uma exigência para todo dispositivo da rede que use o IP, entretanto, muitos dispositivos de segurança, tais como *Firewalls*, obstruem ou incapacitam totalmente ou uma parte da funcionalidade do ICMP com finalidades de segurança.

Como IP provê um serviço de expedição de datagramas sem conexão e não confiável e, além disso, um datagrama trafega de um *gateway* a outro até alcançar um que possa enviá-lo diretamente ao *host* destino, é necessário um mecanismo que emita informações de controle e de erros quando acontecerem problemas na rede. Alguns dos problemas típicos que podem acontecer são:

- Um *gateway* não pode expedir ou direcionar um datagrama
- Um *gateway* detecta uma condição não usual, tal como congestionamento.

O ICMP permite aos *gateways* enviar mensagens de erros ou de controle a outros *gateways* ou *hosts*. Provê comunicação entre os softwares baseados em IP de uma máquina e os de outra máquina.

O ICMP somente reporta condições de erros para a fonte original. A fonte deve relatar os erros aos programas de aplicação individuais e tomar ação para corrigir o problema. Uma das mensagens que o ICMP pode enviar, por exemplo, é: *Destination Unreachable*, o qual, por sua vez pode ser dos seguintes tipos principais:

- *Network Unreachable* (rede não alcançável)
- *Host Unreachable* (host não alcançável)
- *Port Unreachable*[9] (port não alcançável)
- *Destination Host Unknown* (host destino desconhecido)
- *Destination Network Unknown* (rede destino desconhecida)

A ferramenta desenvolvida é capaz de capturar pacotes ICMP e interpretar tipos de mensagens. Desta forma podem ser exibidas mensagens características para cada tipo de pacote ICMP recebido em janelas suspensas, dando transparência ao usuário sobre os erros relatados pelo protocolo. Com isso, espera-se auxiliar na definição não só dos sintomas mas também de alguns sinais disponíveis, e com isso otimizar o suporte.

O ICMP trata várias outras situações além das descritas acima. A tabela 4 ilustra todos os tipos de mensagens ICMP possíveis (POSTEL, 2007).

Tabela 4 - Tipos de Mensagens ICMP

Type	***Mensagem ICMP***
0	Echo Reply
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench
5	Redirect
6	Alternate Host Address
7	Unassigned
8	Echo
9	Router Advertisement
10	Router Solicitation
11	Time Exceeded
12	Parameter Problem
13	Timestamp

[9] As portas tipificam o serviço utilizado. Por exemplo, porta 80: serviço web (HTTP); porta 21: serviço FTP

Type	*Mensagem ICMP*
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
19	Reserved (for Security)
20-29	Reserved (for Robustness Experiment)
30	Traceroute
31	Datagram Conversion Error
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37	Domain Name Request
38	Domain Name Reply
39	SKIP
40	Photuris
41	Experimental mobility protocols
42-255	Reservadas

Alguns tipos de mensagens do ICMP possuem um campo *code*. Este campo particulariza a mensagem de erro (BAKER, 2007). Alguns exemplos extraídos da referência são listados na tabela 5.

Tabela 5 - *Codes* de Mensagens ICMP

Type	*Messagem ICMP*	*Codes*
3	Destination Unreachable	0 Net Unreachable 1 Host Unreachable 2 Protocol Unreachable 3 Port Unreachable 4 Fragmentation Needed and Don't Fragment was Set 5 Source Route Failed 6 Destination Network Unknown 7 Destination Host Unknown 8 Source Host Isolated 9 Communication with Destination Network is Administratively Prohibited 10 Communication with Destination Host is Administratively Prohibited 11 Destination Network Unreachable for Type of Service 12 Destination Host Unreachable for Type of Service 13 Communication Administratively Prohibited

Type	*Messagem ICMP*	*Codes*
		14 Host Precedence Violation 15 Precedence cutoff in effect
5	Redirect	0 Redirect Datagram for the Network (or subnet) 1 Redirect Datagram for the Host 2 Redirect Datagram for the Type of Service and Network 3 Redirect Datagram for the Type of Service and Host
11	Time Exceeded	0 Time to Live exceeded in Transit 1 Fragment Reassembly Time Exceeded
12	Parameter Problem	0 Pointer indicates the error 1 Missing a Required Option 2 Bad Length

Na tabela 5 é possível observar o potencial do ICMP não utilizado pelos Sistemas Operacionais. No caso das mensagens do tipo 3 (*Destination Unreachable*), é possível distinguir se o datagrama enviado não chegou porque não há rota alguma disponível para alcançar o destino, nem sequer uma rota default (código 0); se, apesar da rede desejada ser atingível, o roteador da rede de destino não consegue entregar o datagrama para o endereço de destino final – o mesmo não responde ao ARP (código 1); ou ainda se a comunicação com o endereço de destino final está bloqueada através de filtros administrativos (código 13). As ações de suporte decorrentes são completamente diferentes. Para o primeiro caso, dever-se-ia observar a configuração do *default gateway* da máquina que está originando os datagramas. Possivelmente o mesmo deve estar configurado errado ou não configurado. No segundo caso, apesar de existir a rede cujo DNS apontou na resolução como IP de destino, o roteador desconhece a sua existência. Uma causa possível para isso seria configuração incorreta no DNS *server*, caso se trate de um acesso *web*, ou até mesmo a máquina pode estar inoperante. No terceiro caso, apesar de toda a conectividade física e lógica estar operacional, há uma aplicação, como um *firewall*, bloqueando o acesso. Neste caso, deve-se questionar a legitimidade do acesso, e, caso necessário, verificar a necessidade de desbloquear o acesso no endereço de destino (medida administrativa).Tais informações, apesar de recebidas nos endereços de origem dos datagramas que as ocasionaram, são ignoradas pelos Sistemas Operacionais. Um argumento plausível é que, por tratar-se de um grande número de mensagens em situações de tráfego intenso, a apresentação destas mensagens para o usuário final traria mais transtorno do que soluções, então as mesmas são ignoradas. A opinião do autor é que, se monitoradas de forma discreta e disponível, ainda que na forma de *logs*, as mesmas podem otimizar o suporte e reduzir o tempo de diagnóstico e solução das falhas.

5.3.1.1 Mensagens ICMP tratadas pela Aplicação

Por tratar-se de um protótipo, com finalidade de comprovar a viabilidade da proposta de otimização da atividade de suporte com CBR, foram escolhidas algumas mensagens ICMP consideradas relevantes em relação aos tipos de falhas mais freqüentes:

- **ICMP tipo 0 , Mensagem de pedido de eco**

Esta mensagem é gerada em resposta a uma mensagem ICMP do tipo 8 (mensagem de pedido do eco). O identificador, número de seqüência e os campos de dados devem retornar ao remetente inalterados. Estes dados podem ser usados pelo remetente do eco para ajudar a combinar as respostas com os pedidos do eco, e, caso não haja uma razão para a recepção desta mensagem, há uma grande chance da máquina do usuário estar buscando outras máquinas da rede para disseminação de vírus.

- **ICMP tipo 3, Mensagem de destino inalcançável**

Esta mensagem é gerada por um roteador para informar ao *host* de origem que o endereço de destino é inalcançável. O cabeçalho IP mais os primeiros 8 bytes de dados do datagrama original são retornados ao remetente. Estes dados permitem ao host de origem associar a mensagem recebida ao processo apropriado, e eventualmente detectar uma falha de configuração ou física, por exemplo. O código da mensagem, citado na tabela 5, ajuda a compreender a razão da recepção desta mensagem mais detalhadamente.

- **ICMP tipo 3, Mensagem de destino inalcançável**

Esta mensagem é gerada por um roteador para informar ao *host* de origem que o endereço de destino é inalcançável. O cabeçalho IP mais os primeiros 8 bytes de dados do datagrama original são retornados ao remetente. Estes dados podem ser usados pelo *host* para associar a mensagem ao processo apropriado, e com isso diagnosticar uma falha ocorrida, através do número da porta utilizada. Se, de acordo com a informação nas tabelas de roteamento dos *gateways*, a rede especificada no campo de destino de um datagrama for inalcançável, por exemplo, o cálculo de rota da distância da rede de destino tem resultado infinito, o *gateway* pode emitir uma mensagem de destino inalcançável ao *host* de origem do datagrama, determinando que o destino está inalcançável. Se, no *host* de destino, o módulo IP não puder entregar o datagrama

porque a porta do protocolo ou do processo não está ativa, o *host* do destino pode emitir uma mensagem de destino inalcançável ao *host* de origem. A mensagem de destino inalcançável do ICMP contém um código que descreve a razão pela qual o destino está inalcançável, possibilitando a redução da imprecisão no diagnóstico.

- **ICMP tipo 4, Mensagem de *Source Quench***

Esta mensagem é um pedido para redução da taxa de envio de datagramas. Tipicamente, a mesma ocorre quando se inicia a perda de datagramas por incapacidade do *host* de destino de processar todos os datagramas efetivamente recebidos. O cabeçalho IP acrescido dos primeiros 8 bytes dos dados do datagrama original são retornados ao remetente. Estes dados podem ser usados pelo *host* de origem para combinar a mensagem ao processo causador do problema, e assim reduzir o congestionamento através da redução da taxa de despacho dos datagramas. Um *gateway* (roteador) também pode descartar datagramas se não tiver o espaço de *buffer* (armazenamento) necessário para enfileirar os datagramas para a saída em direção à rede seguinte, na rota para a rede de destino. Se um *gateway* descartar um datagrama, poderá emitir uma mensagem de *Source Quench* ao *host* de origem. O *gateway* pode emitir uma mensagem de *Source Quench* para cada mensagem que descarta. No recebimento de uma mensagem de *Source Quench*, o *host* de origem deve diminuir a taxa em que está enviando o tráfego ao destino até que pare de receber mensagens de *Source Quench*. O *host* de origem pode então gradualmente aumentar a taxa em que emite o tráfego ao destino até que receba outra vez mensagens de *Source Quench*. O *Gateway* ou o *host* podem emitir a mensagem de *Source Quench* quando está se aproximando do limite de armazenamento e esperar até que a capacidade esteja excedida. Isto significa que o datagrama que provocou a mensagem de *Source Quench* poderá ser entregue. Optou-se por monitorar este tipo de mensagem, apesar da reação (redução da taxa de transmissão) ser implementada pelos Sistemas Operacionais, por que a causa desta incapacidade de tratamento dos datagramas pode ser um problema de estreitamento de banda relativo a problemas físicos ou de enlace. Neste caso, a solução correta é o diagnóstico e a resolução da causa do problema, e não apenas a simples redução da taxa, que estará eliminando apenas o sintoma.

- **ICMP, tipo 5, mensagem de redirecionamento**

A mensagem ICMP tipo 5 contém uma informação de redirecionamento para que os datagramas sejam enviados por outra rota. É um mecanismo usado para que roteadores enviem informações de roteamento para os *hosts*.

O endereço IP do *gateway* e do cabeçalho Internet mais os primeiros 8 bytes dos dados do datagrama original são retornados a origem. Estes dados podem ser usados pelo *host* que tentou usar uma rota inadequada para correção do problema. Os códigos 0, 1, 2, e 3 (tabela 5) podem ser recebidos de um *gateway*, contribuindo para melhor esclarecer a razão da ocorrência de uma falha.

- **ICMP TIPO 8 – Mensagens de Pedido de ECO**

Os dados recebidos na mensagem de pedido de eco devem ser devolvidos na mensagem de resposta de eco (tipo 0). O identificador e o número de seqüência podem ser usados pelo remetente do eco para ajudar a combinar as respostas com os pedidos do eco. Esta mensagem foi escolhida para a implementação inicial por permitir a evidenciação de tentativas de verificação de operacionalidade, tipicamente utilizadas por vírus e *hackers*.

- **ICMP tipo 11, mensagem de tempo excedido**

O cabeçalho Internet mais os primeiros 8 bytes de dados do datagrama original são retornados ao remetente. Estes dados podem ser usados pelos *hosts* que recebem a mensagem ICMP tipo 11 para combinar a mensagem ao processo apropriado, ou seja, o que sofreu a falha. Outra situação típica é que a camada IP deve implementar a fragmentação e a remontagem de datagramas IP, de forma a permitir a sua adaptação aos MTU das redes. Este procedimento possui tempos-limite configurados, para evitar a sobrecarga dos equipamentos de conectividade de camada 3 (roteadores). Se este tempo limite expirar, o datagrama parcialmente remontado deve ser rejeitado e uma mensagem de tempo excedido deverá ser emitida ao *host* de origem. A recepção destas mensagens pode indicar não apenas um congestionamento natural da rede, mas, caso esta recepção ocorra freqüentemente em um determinado enlace, pode representar um sintoma de defeitos em interfaces e cabeamento, que provocam o retardo do tratamento do datagrama com o conseqüente estouro do tempo-limite. Outra situação possível é a configuração inadequada deste valor de tempo-limite, inferior ao ideal (ajustado à rede,

com valor default entre 60 e 120 segundos (LOPES *et al.,* 2003), demandando o seu ajuste.

- **ICMP tipo 12, mensagem de problema de parâmetro**

Se um *gateway* ou *host*, ao tentar processar um datagrama, encontrar um problema com os parâmetros do cabeçalho, e por isso não seja capaz de terminar de processá-lo, deverá rejeitar o datagrama. Uma fonte potencial de tal problema é quando há argumentos incorretos em uma opção. Para este tipo de mensagem ICMP, há um ponteiro (4° octeto) que identifica o octeto do cabeçalho do datagrama original onde o erro foi detectado (POSTEL, 2006). O tratamento desta mensagem pode permitir a identificação, por exemplo, Por exemplo, caso o ponteiro indique como razão do problema o *flag* o IP que indica a não-fragmentação obrigatória do datagrama, pode-se supor que o datagrama encontrou em sua rota um MTU inferior ao tamanho do datagrama, o que ocasionaria uma fragmentação. Uma situação como esta demandaria ações de análise da configuração do MTU do segmento responsável pela falha. Nas condições normais, de não observância deste tipo de mensagem pelos Sistemas Operacionais, o emissor continuará tentando enviar o datagrama até que o tempo-limite da espera por datagramas no destino se esgote, e a conexão seja encerrada, ou então a rede consiga enviá-lo por outro caminho, o que ocorre na maioria das vezes.

5.3.1.2 Requisitos da Aplicação Desenvolvida

Visando apenas demonstrar a viabilidade da funcionalidade proposta, foi implementado em linguagem de programação *Delphi* segundo um algoritmo simples, onde a interface de rede é colocada em modo promíscuo, os datagramas ICMP são armazenados em *buffer* e tratados. Apenas os datagramas de interesse são resumidamente evidenciados para o usuário, com a devida análise, e registrados em log. Os demais são descartados para reduzir a sobrecarga do sistema. Sua tela de entrada, apresentada na Figura 30, denota a disponibilidade de três botões de comando (Capturar, Log, Sair), para as suas funções básicas.

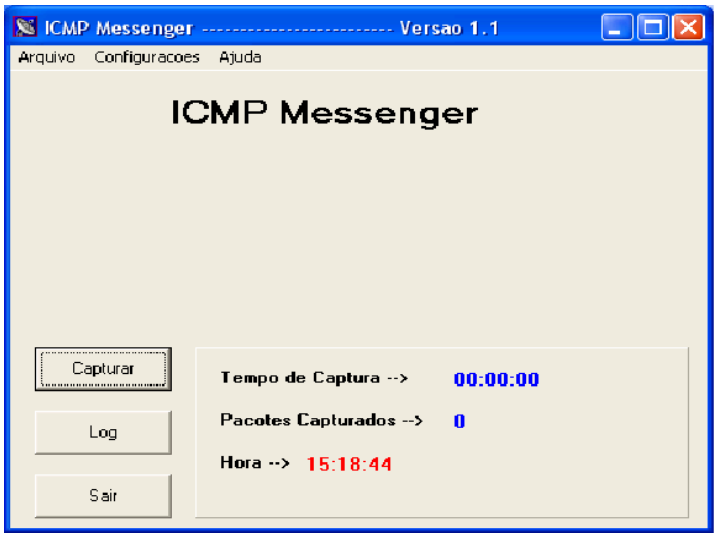


Figura 30 - Aplicação para Tratamento de ICMP

Para o uso adequado da aplicação, inicialmente deve-se ir no menu "Configurações" e no sub-menu "Dispositivos" para se configurar a interface que será utilizada (Figura 31).

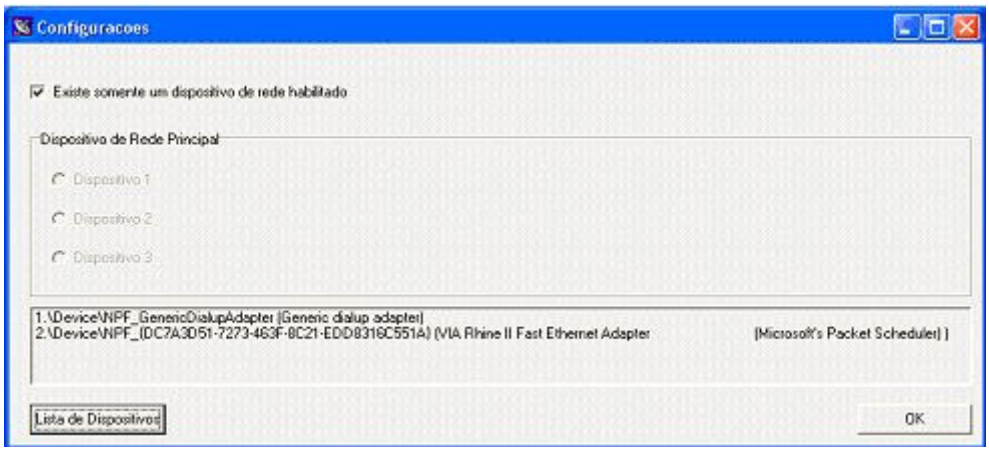


Figura 31 - Configuração da Aplicação

Ao se iniciar a captura dos datagramas, a aplicação opera em segundo plano, alertando ao usuário através de uma janela *pop-up* quando ocorrer a captura de uma mensagem dentre as escolhidas como indispensáveis para o gerenciamento. A intenção é evidenciar possibilidades de falhas em curso, visando atingir a postura pró-ativa com a reação anterior ao evento prestes a ocorrer. A figura 32 ilustra um evento típico de escaneamento de atividade, preliminar aos ataques externos "cegos". Neste tipo de ataque, a ameaça, após constatar a atividade da interface, parte para a segunda fase, do escaneamento de portas. Posteriormente, buscaria vulnerabilidades nas portas abertas, e assim por diante, até conseguir sucesso na invasão do recurso computacional em foco.

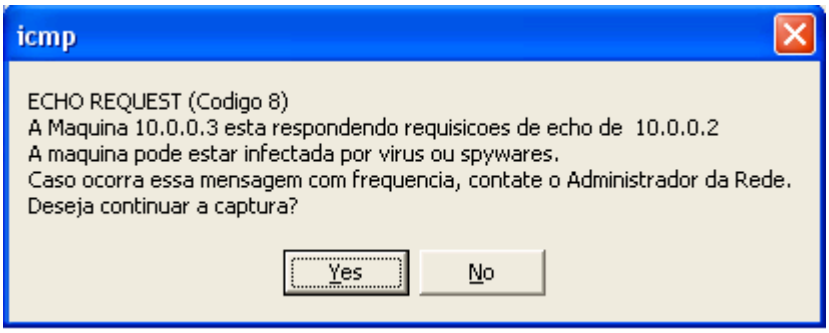


Figura 32 - Informação de Ping

Em alguns casos, um ataque simples de Negação de Serviço, afetando a disponibilidade do recurso computacional, pode ocorrer de forma silenciosa e de difícil detecção do pessoal de suporte. Tipicamente, é perdido algum tempo precioso reiniciando o sistema paralisado e inspecionando situações normais de falha no campo físico, até que se perceba a ocorrência de inundações de requisições para a interface do recurso, mantendo-a ocupada e não conseguindo atender às requisições legítimas. Para reduzir esta latência na reação ao ataque, e possibilitar eventualmente a aplicação de uma contingência antes que a indisponibilidade efetivamente ocorra, foi implementado o tratamento desta situação na sua variante mais simples, ilustrada na figura 33.

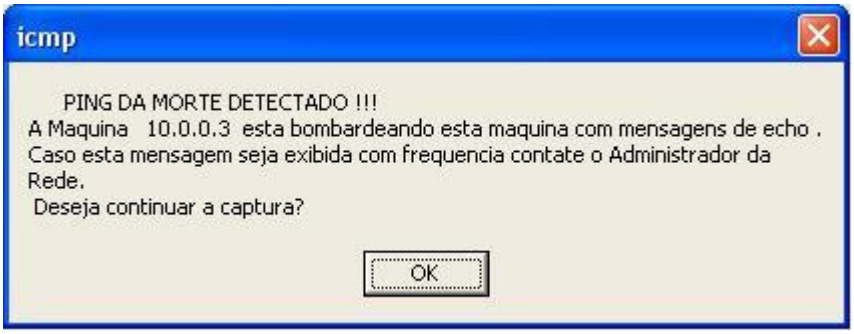


Figura 33 - Inundação de Requisições Detectada

A indisponibilidade de rotas para o alcance do destino é outra situação difícil de ser identificada, já que os navegadores se baseiam em tempos de resposta para apresentar mensagens de "destino inatingível". Imediatamente, a percepção do usuário e até do pessoal do suporte será a de problemas no *host* de destino (pode realmente estar indisponível), ou da presença de alguma falha física que impediu a conectividade. Para isso foi implementada a opção ilustrada na figura 34, mais uma vez contribuindo para melhor definir os sinais característicos da falha.

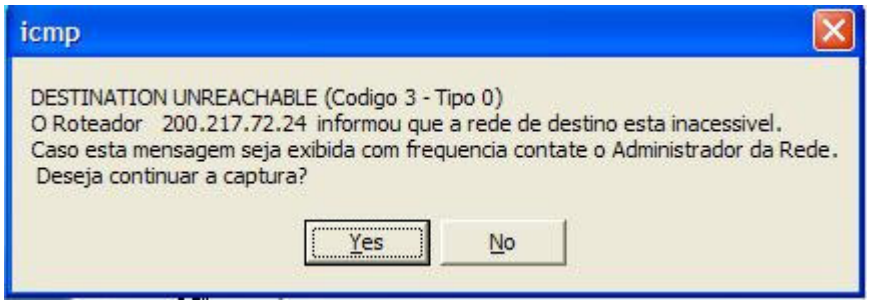


Figura 34 - Tratamento do Destino Inatingível

No caso ilustrado, a combinação código-tipo permitiu um maior nível de conhecimento do evento. Por tratar-se do tipo zero, verifica-se precisamente que não existe uma rota para a rede de destino, ou seja, os cenários de falha possíveis ficam limitados a um conjunto bem menor de possibilidades, eliminando, por exemplo, a possibilidade de falhas físicas entre a máquina de origem e o roteador citado, evento que certamente precisaria ser testado caso não se tivesse conhecimento desta mensagem.

Todas as mensagens tratadas são armazenadas em *log*, que pode ser visualizado através do acionamento da opção *log* na tela principal. A figura 35 ilustra esta funcionalidade.

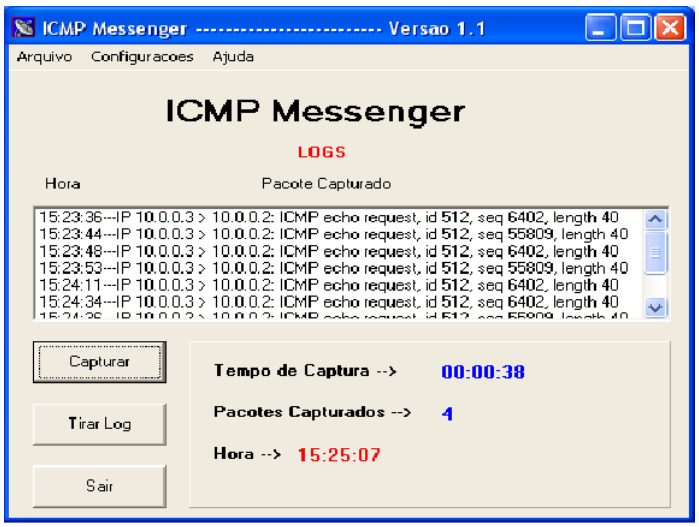


Figura 35 - Log de Mensagens ICMP Tratadas

As oito mensagens descritas na seção 7.1.1 foram implementadas, e os resultados de sua aplicação no *test-bed* são descritos no capítulo dos resultados.

O uso das mensagens de ICMP como ferramenta de auxílio à percepção de sintomas e sinais poderá otimizar de forma significativa a tarefa de diagnosticar falhas em redes. Como toda funcionalidade que aumenta a segurança, a performance é reduzida devido à captura dos datagramas para filtragem dos que sejam ICMP e seu posterior processamento. Uma possibilidade razoável para eliminar isso é a instalação

de máquinas dedicadas apenas ao gerenciamento em portas de dispositivos de conectividade configuradas como SPAN[10], funcionalidade tipicamente disponível em dispositivos gerenciáveis. Aperfeiçoamentos ao programa também podem ser feitos para operação apenas quando houver disponibilidade do sistema, ou quando uma nova conexão for tentada.

A ferramenta descrita a seguir complementa esta funcionalidade com a leitura de arquivos de configuração e testes primários, com o objetivo de permitir, de forma simples, objetiva e prática, dados básicos necessários para a análise preliminar de sintomas descritos pelos usuários. A idéia é manter a abstração do usuário quanto aos detalhes da sua conectividade, mas possibilitar o acesso transparente a informações fundamentais para o diagnóstico que apenas poderiam ser obtidas através de ferramentas de rede complexas para a grande maioria dos usuários finais. Como não é necessária sobrecarga alguma ao sistema para sua disponibilização, a mesma pode perfeitamente ser inserida no contexto operacional das estações, contribuindo para melhoria da percepção do usuário quanto aos sintomas das falhas.

5.3.2 Aplicação para Descrição do Ambiente de Rede

Voltados para dar transparência ao usuário dos detalhes mais técnicos, os Sistemas Operacionais não possuem caminhos simples para o tratamento de falhas de conectividade. Os *wizards* de configuração são capazes de lidar apenas com situações padronizadas e configurações genéricas, sugerindo sempre ao usuário que busque auxílio com o administrador da rede. Ocorre que, com a popularização do acesso em banda larga, aliado aos modismos do mundo moderno, com os *chats*, *blogs*, comunidades, *videologs* e *fotologs*, além da imensa disponibilidade de material de pesquisa, é cada vez mais comum a criação de redes domésticas. Nestas redes, não há a figura do administrador, e as operadoras de banda larga não se responsabilizam pelo suporte de redes domésticas. Neste cenário, seria interessante um investimento na transparência destes aspectos, facilitando a percepção de sinais de falhas com a obtenção de informações sintetizadas sobre a sua conectividade, de forma que as mesmas possam, ao menos, serem comparadas com as configurações esperadas, e assim possibilitar um diagnóstico mais rápido e eficaz. No caso do ambiente de teste utilizado neste trabalho, o objetivo é permitir o enriquecimento da descrição de um caso de falha, encurtando o tempo para solução da mesma.

[10] SPAN – *Switched Port Analyzer* – Porta de comutador em modo permanentemente promíscuo.

Para atingir este objetivo, foi desenvolvido um aplicativo para permitir, ao administrador da rede, testar remotamente a configuração de uma determinada estação de trabalho. Uma vez que o ambiente de teste é híbrido, com estações de trabalho executando os Sistemas Operacionais (SO) Windows XP, 2000 ou 2003, e várias distribuições de Linux, como o *QNX*, *Debian*, *Suse*, *Ubuntu* e *Kurumim*, optou-se por desenvolvê-lo em Java e torná-lo transparente ao SO utilizado. O comando para verificação das configurações de interfaces, por exemplo, alterna entre *ipconfig* e *ifconfig* automaticamente, mediante a verificação pelo cliente de qual SO está instalado na estação de trabalho onde está sendo executado o servidor.

O diagrama de classes para o aplicativo é ilustrado na figura 36.

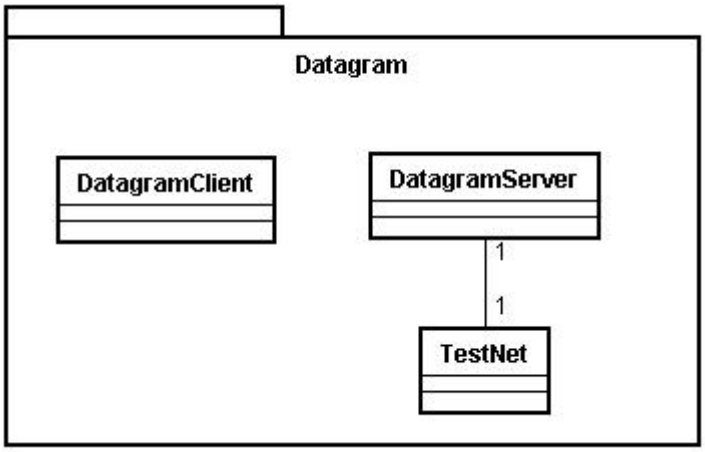


Figura 36 - Diagrama de Classes do Programa de Descoberta de Informações de Rede

No diagrama da figura 36 são apresentadas as três classes:

- ***DatagramClient*** – classe cliente;
- ***DatagramServer*** – classe servidor; e
- ***TestNet*** – classe de teste da configuração.

Para a comunicação entre o servidor e o cliente é empregado o protocolo UDP, uma vez que não é necessário estabelecer um canal confiável entre o cliente e o servidor. Esta opção contribui para reduzir o custo computacional do uso da ferramenta, e a perda de uma mensagem UDP de requisição ou resposta pode ser resolvida enviando-se uma nova solicitação pelo cliente.

O processo de comunicação consiste de um programa cliente e diversos servidores, um em cada estação de trabalho, conforme a ilustração da figura 37.

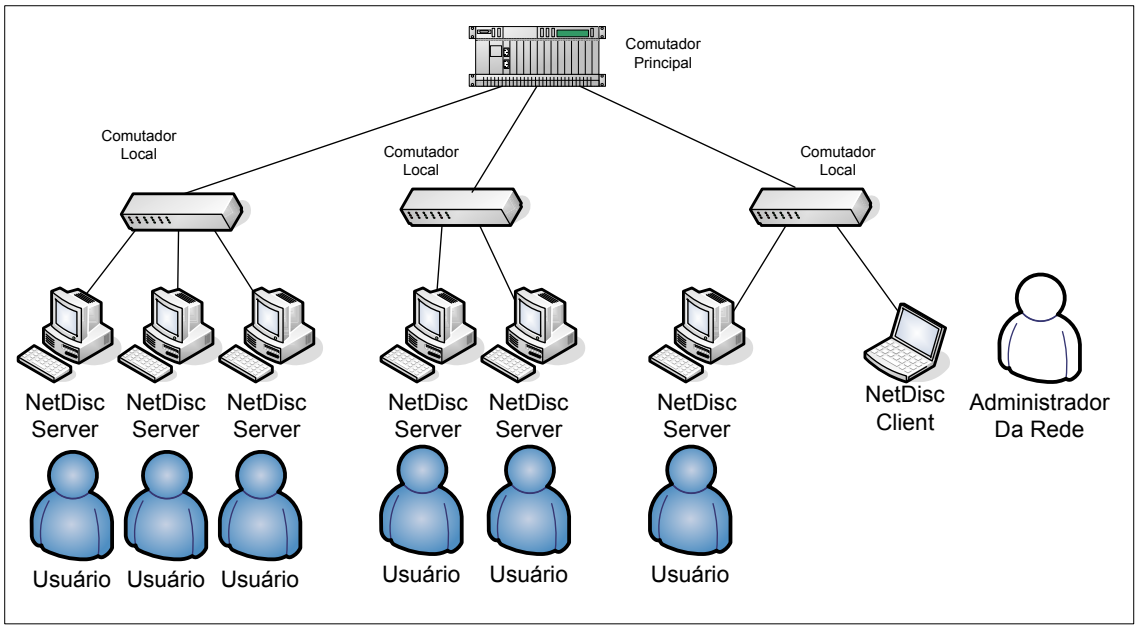


Figura 37 - Ambiente de Operação do Netdisc

Quando há a necessidade de verificação de alguma estação de trabalho, o programa *NetDisc* cliente tenta estabelecer comunicação com o servidor instalado nesta estação e o interroga sobre a sua configuração.

- **Classe DatagramClient**

  A classe *DatagramClient* implementa o programa cliente.

  Para a comunicação via UDP emprega o pacote java.net e utiliza as classes:

  - ✓ **DatagramPacket** – que representa o pacote de dados enviados ou recebidos; e
  - ✓ **DatagramSocket** – que representa a conexão por onde tramitam os pacotes.

  O cliente possui dois comandos implementados:

  - ✓ **ifconfig** – que espera como resposta a saída do comando "*ipconfig*", para Windows XP/2000, e do comando "*ifconfig*" para distribuições Linux. Ao receber o pacote de retorno, o apresenta na tela.
  - ✓ **Bye** – termina o programa.

- **Classe DatagramServer**

  Esta classe também emprega as mesmas classes do pacote *java.net*, porém fica aguardando a entrada de algum pacote para responder.

Ao receber o comando "*ifconfig*", o aplicativo chama o programa correspondente, de acordo com o SO da estação de trabalho, que retorna a sua saída através de uma mensagem UDP.

Para rodar o comando emprega a classe *TestNet*.

- ✓ **TestNet –** Esta classe é responsável pela identificação dos parâmetros da estação de trabalho. Para atingir este intuito esta classe emprega as seguintes classes do pacote *java.net*:
  - *InetAddress* - Por meio desta classe é obtido o endereço associado a cada interface instalada; e
  - *NetworkInterface* – Por meio desta classe são identificadas as interfaces instaladas na estação de trabalho.

A classe *TestNet* também possui o método *ping* que pode ser empregado para verificar a operacionalidade de outros endereços.

O método *ifconfig* da classe *TestNet* é responsável por chamar o programa "*ipconfig*", no Windows, ou o programa "*ifconfig*", no ambiente Linux. Este método retorna uma *string* com a saída destes comandos.

A partir dos dados retornados pelo método *ifconfig*, a classe *TestNet* identifica a máscara empregada, o endereço dos servidores DNS configurados, e confirma o endereço das interface.

Para executar o programa cliente deve ser usado o comando:

```
java –classpath cliente.jar datagram.DatagramClient <IP do
servidor> 1234
```

Onde "1234" foi a porta escolhida para o *socket* de comunicação no servidor[11].

Para rodar o programa servidor deve ser usado o comando:

```
java –classpath server.jar datagram.DatagramServer
```

Como é o programa servidor que permanece instalado nas estações, em casos onde não haja conectividade com a máquina do usuário para obtenção das informações, pode-se solicitar ao próprio que execute os comandos da classe *TestNet* e diga por telefone os resultados obtidos, evitando assim um deslocamento do pessoal de suporte até o ambiente do cliente. Como será apresentado nos resultados, no capítulo 7, o uso do aplicativo, batizado provisoriamente de *NetDisc*, possibilitou grande redução no tempo de paralisação das estações por erros triviais de configuração.

[11] Nota do autor: apesar da porta 1234 UDP estar vinculada a um serviço de busca (*Infoseek*), é possível que o tráfego seja interpretado por algum antivírus como ameaça, já que a porta TCP 1234 é usada pelo *Subseven Java client* e pelo *Ultors trojan*.

As figuras a seguir ilustram a utilização do protótipo do *NetDisc*. Na figura 38, o programa servidor é executado em uma estação de trabalho da rede. Na figura 39, o programa cliente, executado na máquina do administrador da rede, obtém as informações da estação de trabalho em estado falha, para melhor descrição do caso.

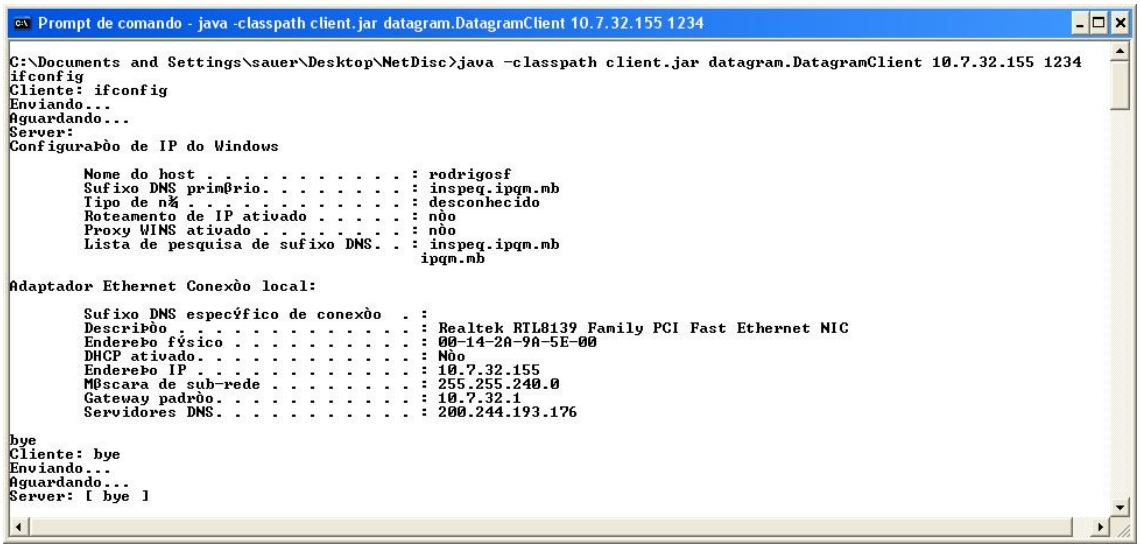


Figura 38 - Tela da aplicação cliente, executada pelo Administrador da Rede

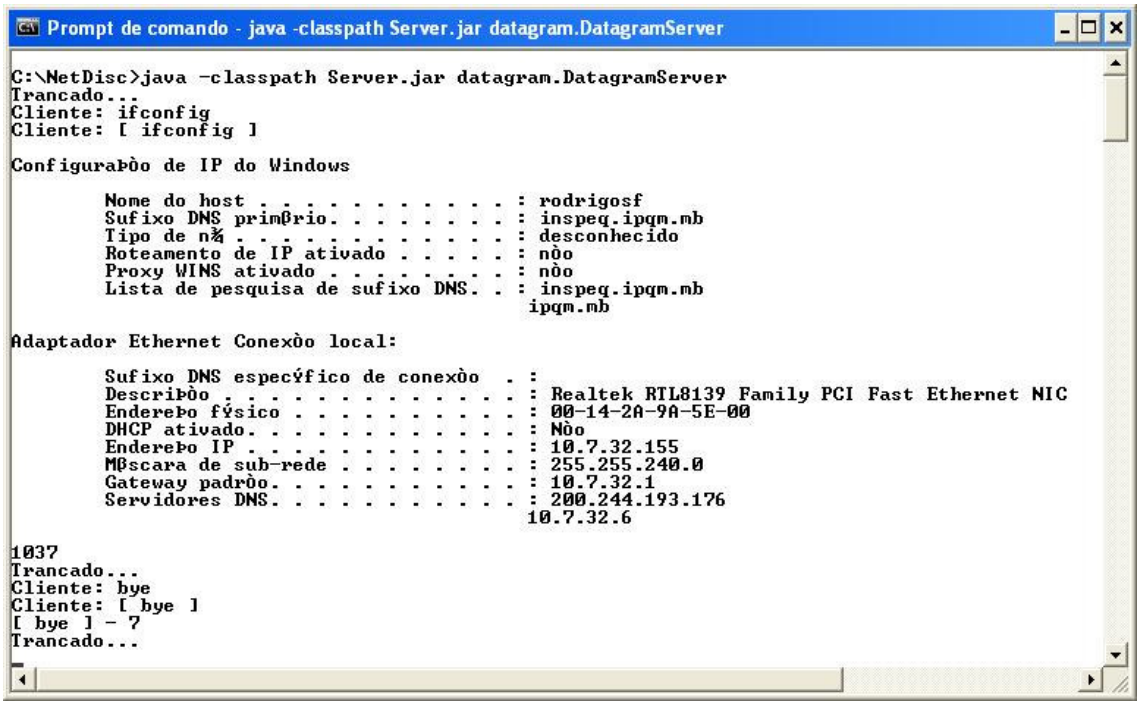


Figura 39 - Tela da Aplicação Servidor, instalada nas estações da Rede Local

As figuras 38 e 39 ilustram o funcionamento do protótipo. A aplicação “servidor” fica instalada nas estações da Rede Local, disponibilizando a porta UDP 1234 para envio de suas informações de rede. O Administrador da Rede Local possui em sua estação a aplicação cliente. Através do endereço IP da máquina em condição de falha, o

Administrador executa o comando ilustrado na figura 38, estabelecendo assim um canal de comunicação entre as máquinas. Ao digitar *ifconfig*, caso o SO da estação reclamante seja Windows, é executado o procedimento correspondente ao comando *ipconfig*. Se o SO for Linux, é executado o comando *ifconfig*. A saída do comando na estação em falha é capturada pela aplicação e enviada para a aplicação cliente, possibilitando assim que importantes informações sejam utilizadas na descrição do caso.

Uma nova versão do aplicativo encontra-se em desenvolvimento, visando acrescentar ao protótipo as seguintes funcionalidades:

- Substituição dos complexos comandos em Java por ícones no lado do cliente (Administrador da Rede);
- Incorporação da aplicação Servidor no ambiente operacional dos usuários, na inicialização das estações de trabalho;
- Implementação da consulta local destas informações através de um ícone na área de trabalho, para o caso de não haver conectividade da estação de trabalho do usuário com o Administrador, por exemplo, por problemas em equipamentos de conectividade no caminho entre eles; e
- Um mecanismo de segurança para garantir que esta aplicação não se torne uma vulnerabilidade passível de exploração das máquinas dos usuários por ameaças.

Neste capítulo foram concentradas as principais contribuições deste trabalho. Foi descrita a estratégia de implementação da aplicação CBR para a atividade de diagnóstico e correção de falhas de Redes de Computadores. No contexto desta estratégia, adotou-se a metodologia INRECA, hoje adotada pela maioria dos sistemas CBR apresentados na literatura, visando o maior nível de conformidade possível com o estado-da-arte e a possibilidade da continuidade do trabalho por outros pesquisadores. Foram apresentadas as duas ferramentas desenvolvidas para reduzir o tempo de indisponibilidade das estações de trabalho, através do enriquecimento da descrição dos casos. Ambas mostraram sua utilidade, uma vez que demonstraram atingir este objetivo conforme será apresentado no capítulo 7. O próximo capítulo descreve Estudos de Caso de situações de falha típicas, solucionadas através da aplicação CBR desenvolvida neste trabalho.

6 Estudos de Caso

Visando ilustrar a aplicação da metodologia de otimização do diagnóstico de falhas em redes, serão descritas três situações reais ocorridas no *test-bed* utilizado, durante o tempo de pesquisa de tese:

- ✓ Contaminação por vírus (falha de aplicação);
- ✓ Endereçamento incorreto (falha de rede); e
- ✓ Porta de *switch* com defeito (falha física).

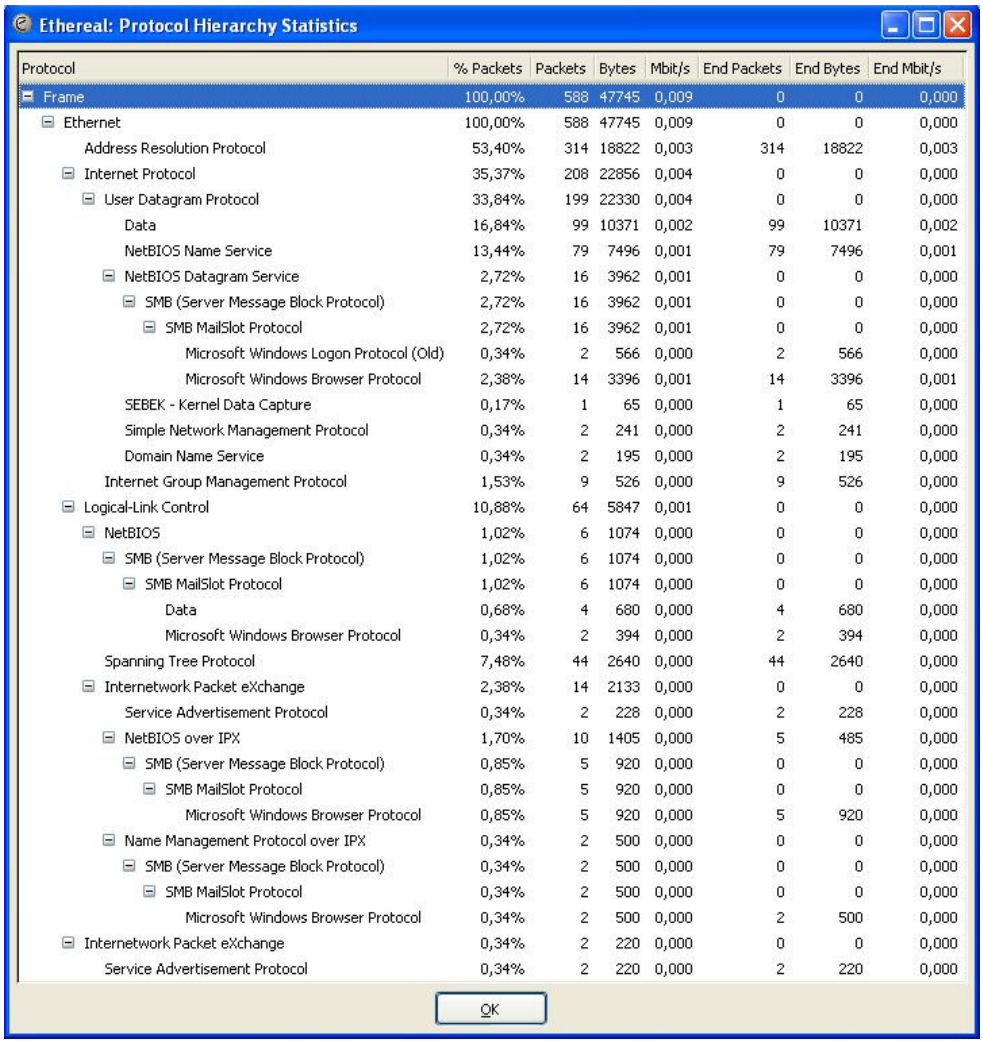
6.1 Caso 1 – Contaminação Viral

Hoje uma das maiores aflições dos Gerentes de Redes, as contaminações virais são de difícil controle por prescindir da disciplina do usuário final, o que nem sempre ocorre. Como os vírus e suas variantes são propagados através de técnicas que podem, no mínimo, ter seu potencial de replicação bastante reduzido em uma rede, é conveniente o seu tratamento rápido, antes que os efeitos provoquem perda da produtividade da atividade-fim do ambiente onde a rede está instalada.

O vírus é um programa que se propaga tipicamente "infeccionando" inicialmente arquivos ou áreas do Sistema Operacional de uma estação de trabalho, fazendo cópias de si mesmo (ANÔNIMO, 2007). Alguns vírus são destrutivos, eliminando ou corrompendo arquivos, e a maioria deles apenas causa lentidão exponencial da rede, conforme a sua propagação é bem sucedida. Atualmente, a forma mais comum de propagação dos vírus é o compartilhamento de arquivos através de dispositivos portáteis de armazenamento, e de anexos de correio eletrônico contaminados. Diferentemente dos *worms*, caso descrito nesta seção, os vírus requerem ações do usuário para que a contaminação ocorra, sendo assim mais passíveis de controle pela educação do usuário. Os *worms*, por sua vez, não dependem da ação do usuário. São desenvolvidos para explorar vulnerabilidades nos Sistemas Operacionais que permitam a transmissão de arquivos sem consulta ao usuário através da rede. Devido ao seu *modus-operandi*, os *worms* são mais rapidamente percebidos pelos usuários em uma rede, porque mesmo os usuários de máquinas protegidas e não contaminadas perceberão lentidão, devido ao consumo de banda pelas máquinas contaminadas na tentativa de propagarem suas infecções.

Durante os testes da arquitetura proposta, uma excessiva lentidão na rede foi percebida por um grande número de usuários. Em busca dos sinais diferenciais que

enriquecem um caso, foi imediatamente acionado um *sniffer* de uso gratuito e de código aberto, o Ethereal (ANÔNIMO, 2006). A captura de 30 segundos de tráfego evidenciou, mesmo antes da apuração do sumário de tráfego, uma taxa bastante alta de *broadcasts*. Esse sinal pode ser constatado na consulta do resultado sintético da captura, ilustrado na figura 40.



Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100,00%	588	47745	0,009	0	0	0,000
Ethernet	100,00%	588	47745	0,009	0	0	0,000
Address Resolution Protocol	53,40%	314	18822	0,003	314	18822	0,003
Internet Protocol	35,37%	208	22856	0,004	0	0	0,000
User Datagram Protocol	33,84%	199	22330	0,004	0	0	0,000
Data	16,84%	99	10371	0,002	99	10371	0,002
NetBIOS Name Service	13,44%	79	7496	0,001	79	7496	0,001
NetBIOS Datagram Service	2,72%	16	3962	0,001	0	0	0,000
SMB (Server Message Block Protocol)	2,72%	16	3962	0,001	0	0	0,000
SMB MailSlot Protocol	2,72%	16	3962	0,001	0	0	0,000
Microsoft Windows Logon Protocol (Old)	0,34%	2	566	0,000	2	566	0,000
Microsoft Windows Browser Protocol	2,38%	14	3396	0,001	14	3396	0,001
SEBEK - Kernel Data Capture	0,17%	1	65	0,000	1	65	0,000
Simple Network Management Protocol	0,34%	2	241	0,000	2	241	0,000
Domain Name Service	0,34%	2	195	0,000	2	195	0,000
Internet Group Management Protocol	1,53%	9	526	0,000	9	526	0,000
Logical-Link Control	10,88%	64	5847	0,001	0	0	0,000
NetBIOS	1,02%	6	1074	0,000	0	0	0,000
SMB (Server Message Block Protocol)	1,02%	6	1074	0,000	0	0	0,000
SMB MailSlot Protocol	1,02%	6	1074	0,000	0	0	0,000
Data	0,68%	4	680	0,000	4	680	0,000
Microsoft Windows Browser Protocol	0,34%	2	394	0,000	2	394	0,000
Spanning Tree Protocol	7,48%	44	2640	0,000	44	2640	0,000
Internetwork Packet eXchange	2,38%	14	2133	0,000	0	0	0,000
Service Advertisement Protocol	0,34%	2	228	0,000	2	228	0,000
NetBIOS over IPX	1,70%	10	1405	0,000	5	485	0,000
SMB (Server Message Block Protocol)	0,85%	5	920	0,000	0	0	0,000
SMB MailSlot Protocol	0,85%	5	920	0,000	0	0	0,000
Microsoft Windows Browser Protocol	0,85%	5	920	0,000	5	920	0,000
Name Management Protocol over IPX	0,34%	2	500	0,000	0	0	0,000
SMB (Server Message Block Protocol)	0,34%	2	500	0,000	0	0	0,000
SMB MailSlot Protocol	0,34%	2	500	0,000	0	0	0,000
Microsoft Windows Browser Protocol	0,34%	2	500	0,000	2	500	0,000
Internetwork Packet eXchange	0,34%	2	220	0,000	0	0	0,000
Service Advertisement Protocol	0,34%	2	220	0,000	2	220	0,000

Figura 40 - Estatísticas de Tráfego

Na figura 40 é possível perceber o tráfego ARP representando mais da metade do tráfego total (53,40%). A submissão dos sintomas à aplicação disponível no CBR-Works apresentou o resultado ilustrado na figura 41.

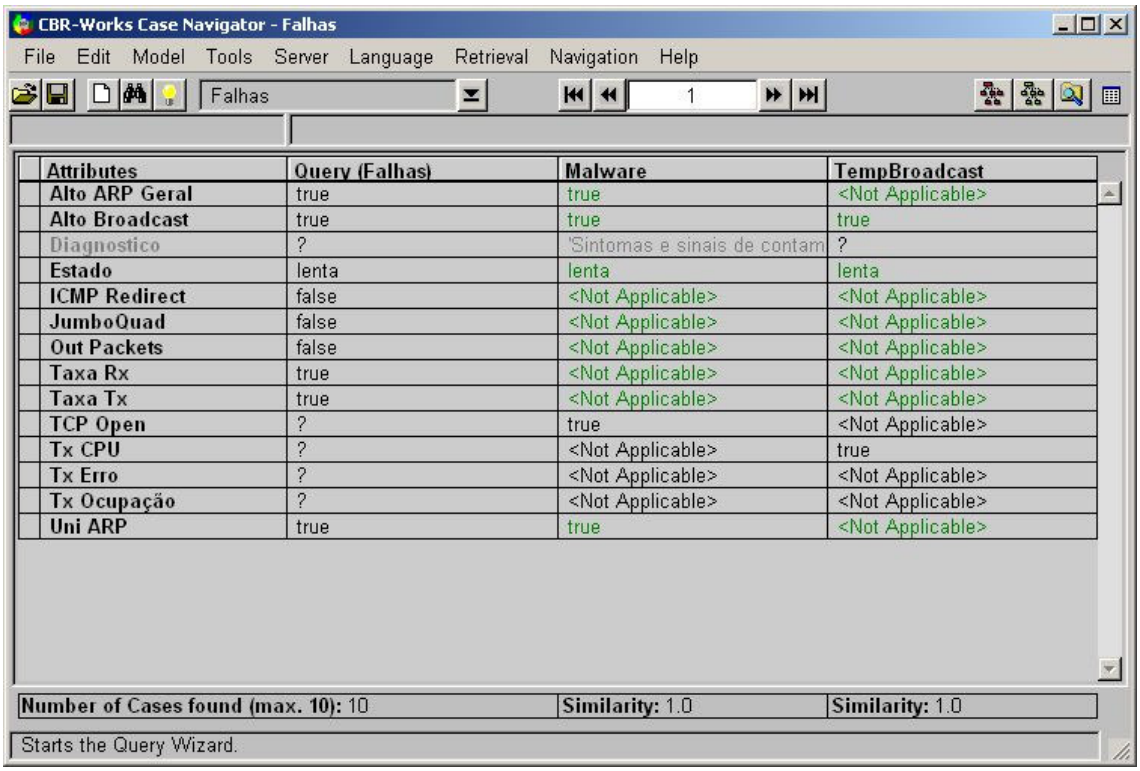


Figura 41 - *Query* de Sintomas e Sinais

Em função do resultado obtido, a aplicação indicou como melhor caso o "*Malware*", situação cujo diagnóstico sugeria o procedimento ilustrado na figura 42:

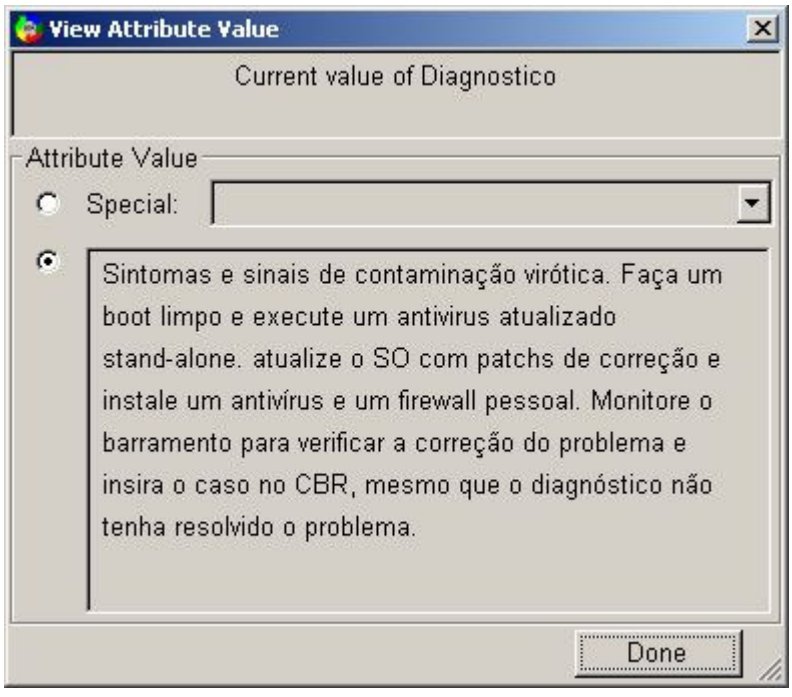


Figura 42- Diagnóstico de Malware

O procedimento indicado demanda a identificação da máquina contaminada. Para isso, uma análise mais detalhada do tráfego possibilitou a identificação de uma máquina com indícios de contaminação viral, conforme ilustrado na figura 43.

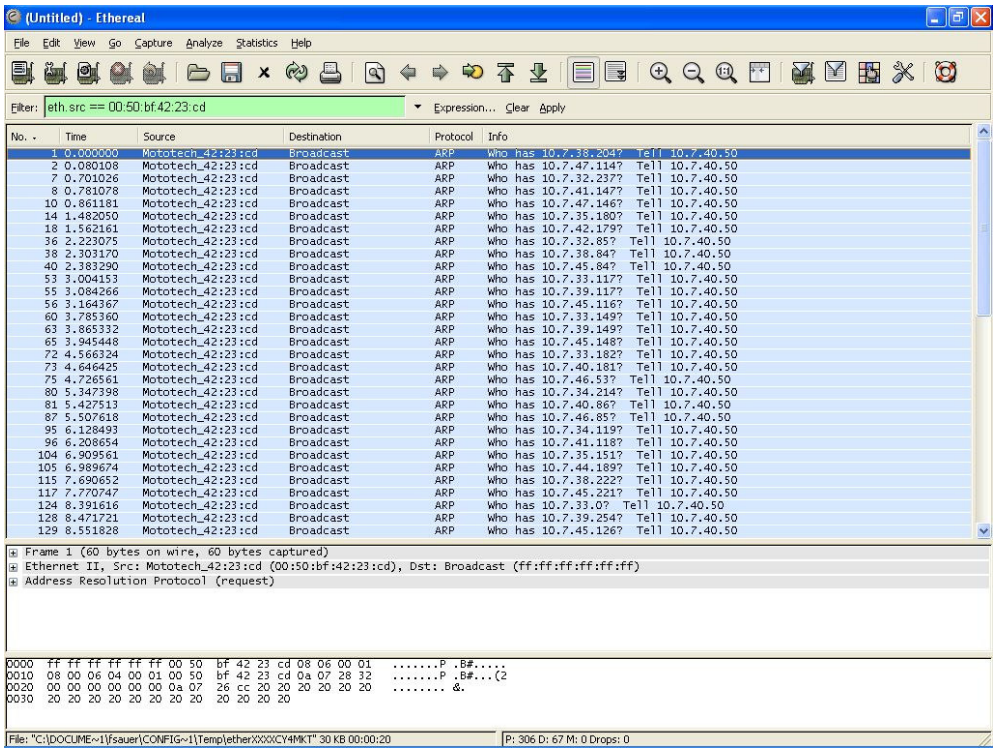


Figura 43 - Broadcasts ARP filtrados

A identificação da máquina permitiu a sua análise local. O único parâmetro ainda indefinido na consulta ao CBR (figura 41), portas TCP abertas, foi confirmado, como ilustrado na figura 44.

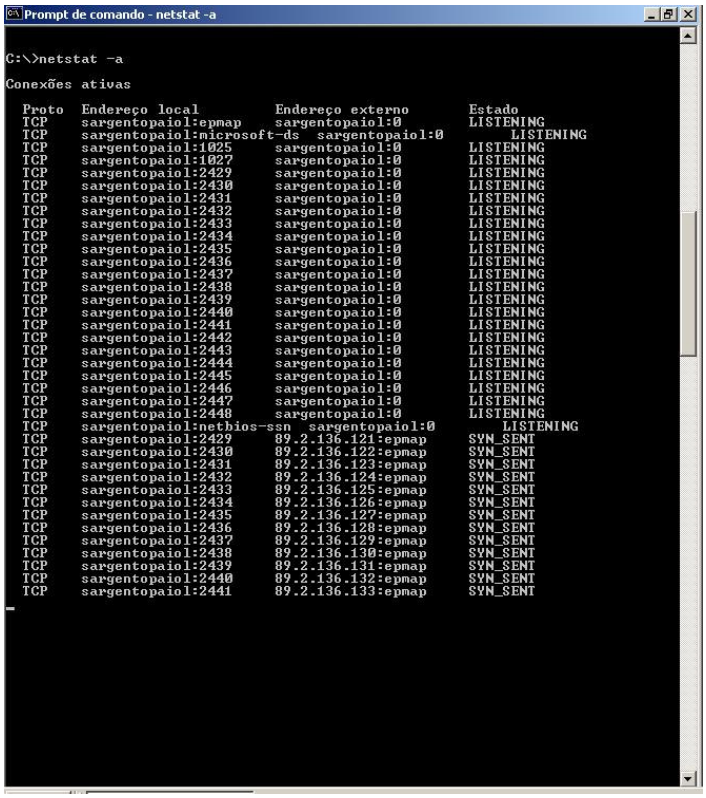


Figura 44 - Portas TCP abertas

Ao inspecionar os processos ativos, verificou-se a presença de uma aplicação estranha ao ambiente operacional típico, conforme pode ser observado na figura 45.

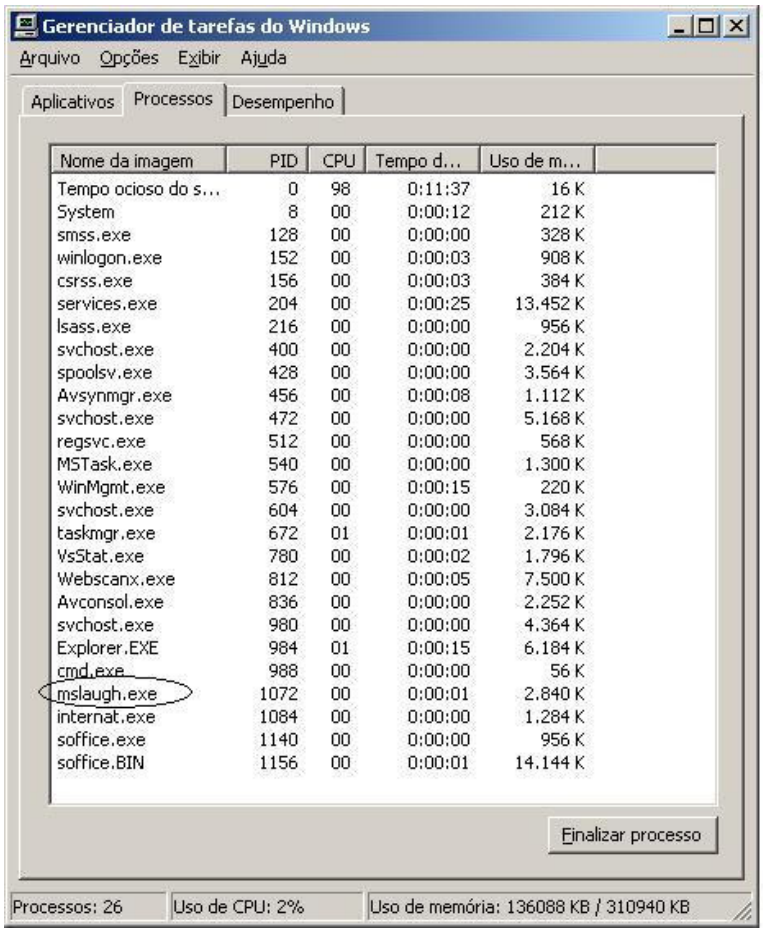


Figura 45 - Processos em Execução

Uma rápida pesquisa na Internet permitiu a constatação de tratar-se de um *worm* documentado, o *mslaugh*. Sua descrição pode ser vista na figura 46.

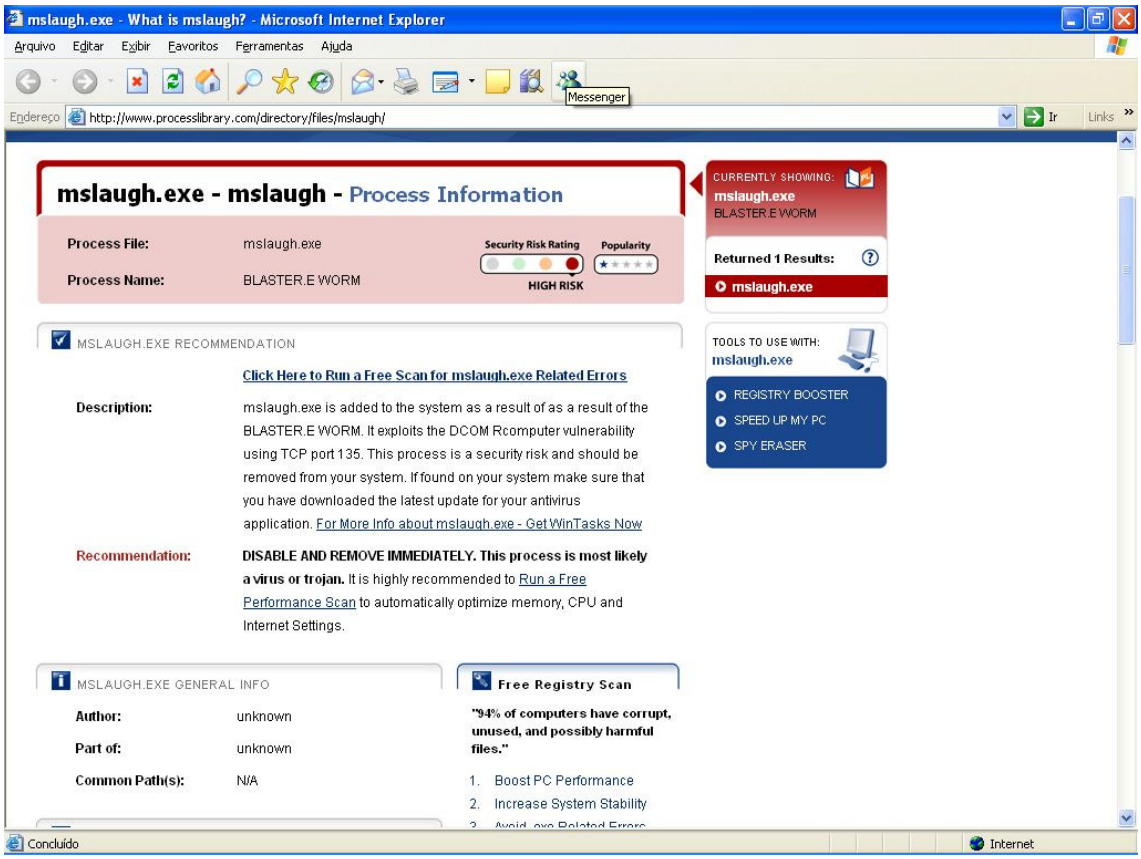


Figura 46 - Descrição do *Worm* MSLAUGH

Com a sua remoção, realizada de acordo com o diagnóstico obtido no *CBR-Works*, a rede voltou a sua condição normal, como pode ser verificado na figura 47.

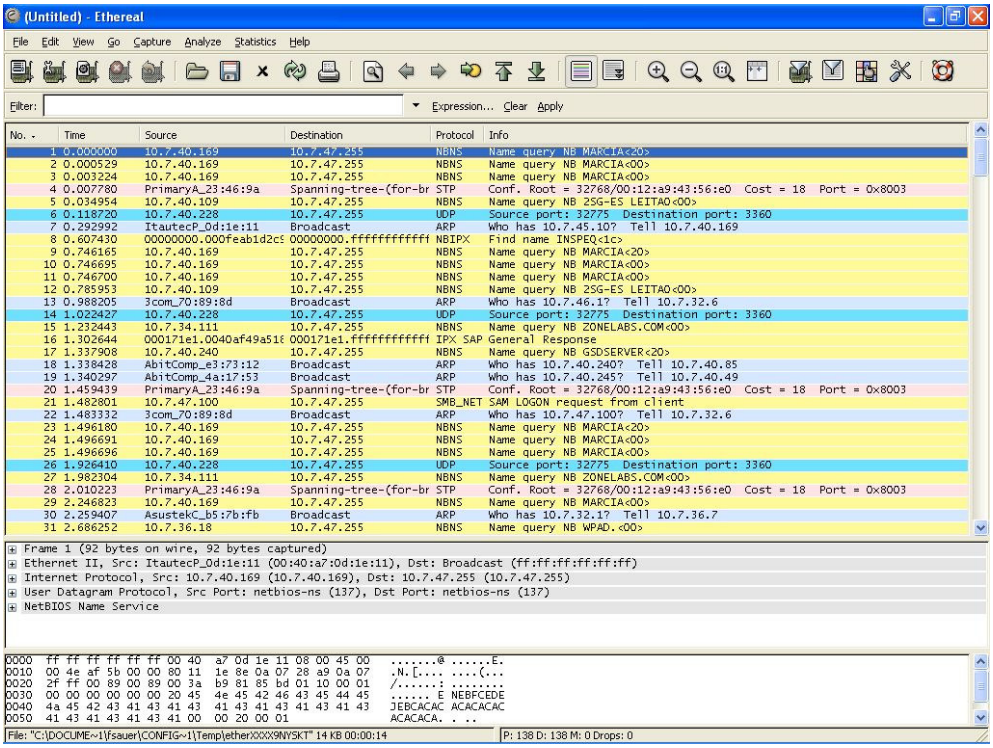


Figura 47 - Tráfego de Rede Normal

6.2 Caso 2 – Endereçamento Incorreto

Para ilustrar os efeitos de erros no endereçamento, serão apresentados nesta seção dois casos típicos: erros na digitação da máscara de subrede com mais ou menos bits no prefixo, reduzindo ou aumentando, respectivamente, o escopo de alcance da Rede Local. Este tipo de erro é freqüente porque, em ambos os casos, a conexão fica intermitente, de acordo com o endereço buscado. A primeira ilustração é a da máscara menor, ou seja, reduzindo o número de bits do prefixo. Neste caso, a estação modifica a sua visibilidade da Rede, entendendo que endereços além dos limites de seu alcance direto (comunicação na camada de enlace) podem ser obtidos via broadcast. O problema disso é que o endereço de broadcast calculado nessa situação errática é diferente do real, fazendo que as estações não recebam esta solicitação e, consequentemente, não respondam à máquina solicitante com seus respectivos endereços de enlace. O *DNS Server*, no entanto, estando corretamente configurado, responderá a esta máquina pelas solicitações de acesso aos servidores de aplicações e *web servers* e, caso os mesmos possuam endereços dentro do escopo de rede configurado, a conexão obterá sucesso sem a emissão de mensagens de erro na rede. Isto confere uma característica intermitente para a falha. As figuras a seguir ilustram a situação descrita.

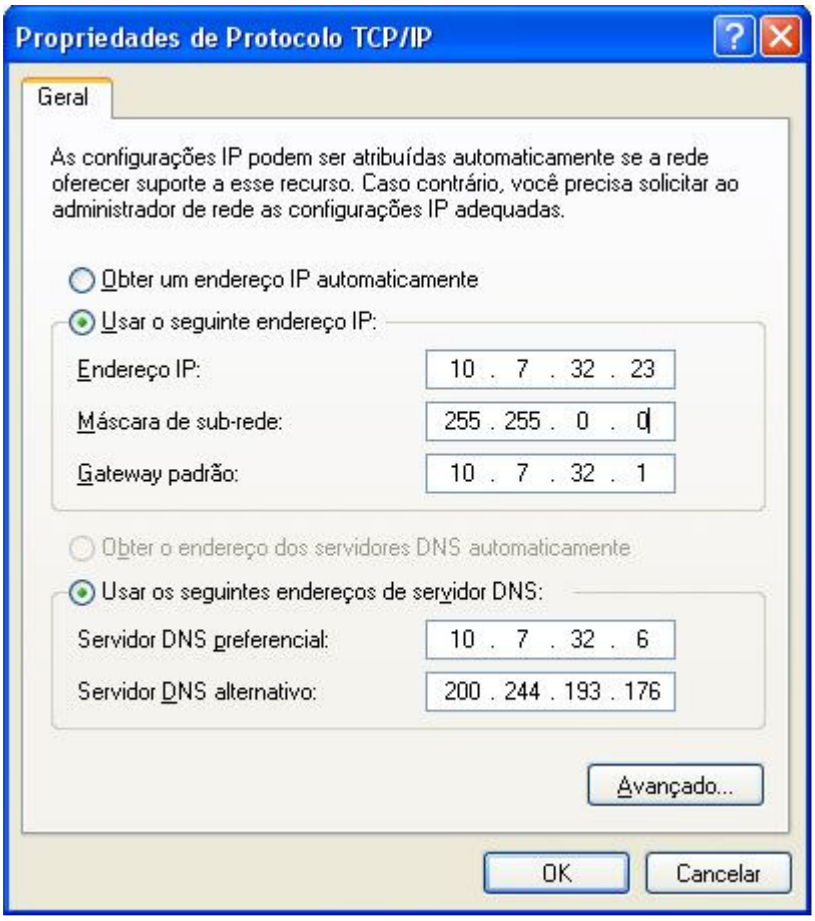


Figura 48 - Endereço com máscara menor

A figura 48 ilustra um erro na digitação da máscara, cuja numeração correta seria 255.255.240.0. Com isto, endereços como o 10.7.48.1, não pertencente à rede, será visto como um endereço acessível através de uma comunicação no nível de enlace, não dependendo de *gateway* para tal. Nesta situação, a principal reclamação é a seguinte: "*consigo acessar os serviços principais da rede, mas não consigo enxergar o meu ambiente de rede*". Para ilustrar esta falha, após a alteração da máscara, conforme apresentado na figura 48, foi esvaziada a tabela ARP da estação de teste e capturado o tráfego. Conforme era esperado, a alteração da máscara provocou o cálculo equivocado do endereço de *broadcast*. Nesta situação, o broadcast da rede seria 10.7.255.255, e não 10.7.47.255, com a máscara correta. O efeito disso é ilustrado na figura 49.

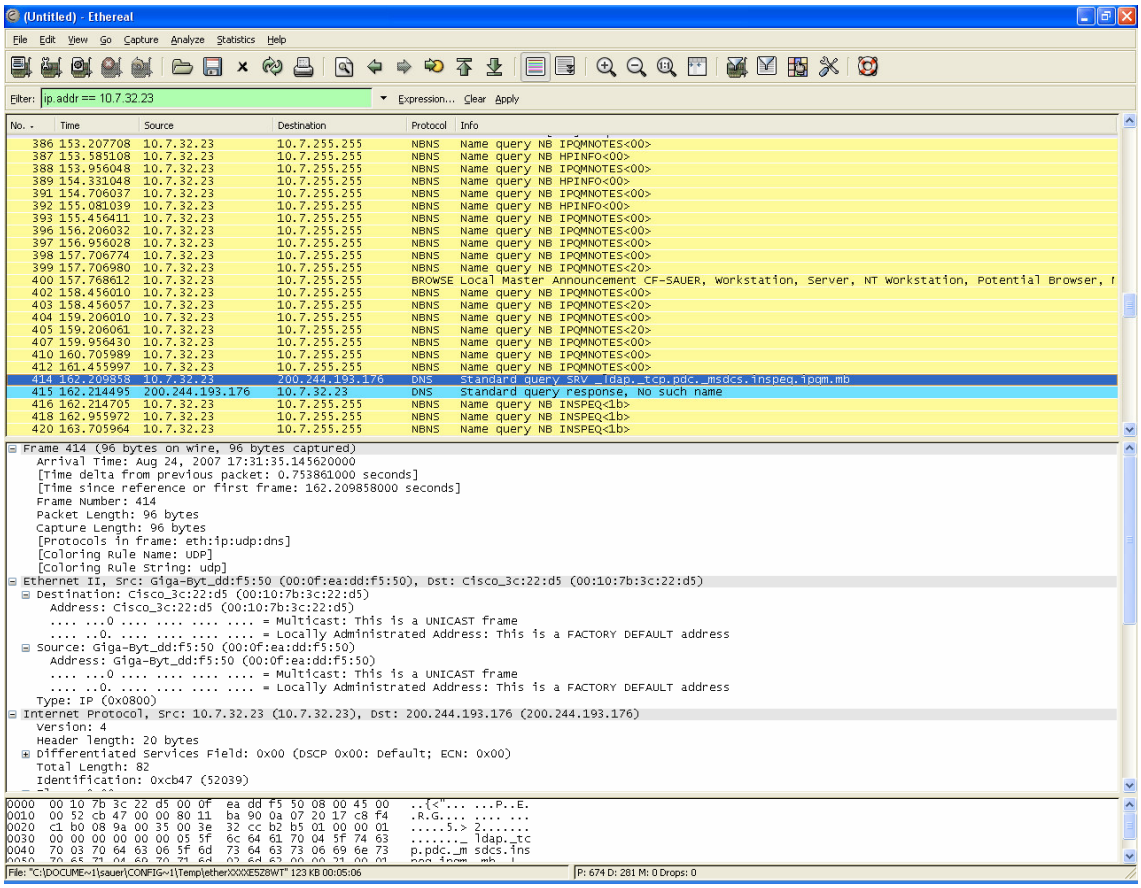


Figura 49 - Procura de estações com *broadcast* errado

A captura ilustrada na figura 49 é representativa da situação de falha. A máquina permanece buscando comunicações para "montar" sua visão da rede, porém através do endereço de *broadcast* errado. Além de aumentar o tráfego da rede tornando-a lenta, o usuário não consegue visualizar o seu ambiente de rede. A figura 50 mostra o cenário descrito.

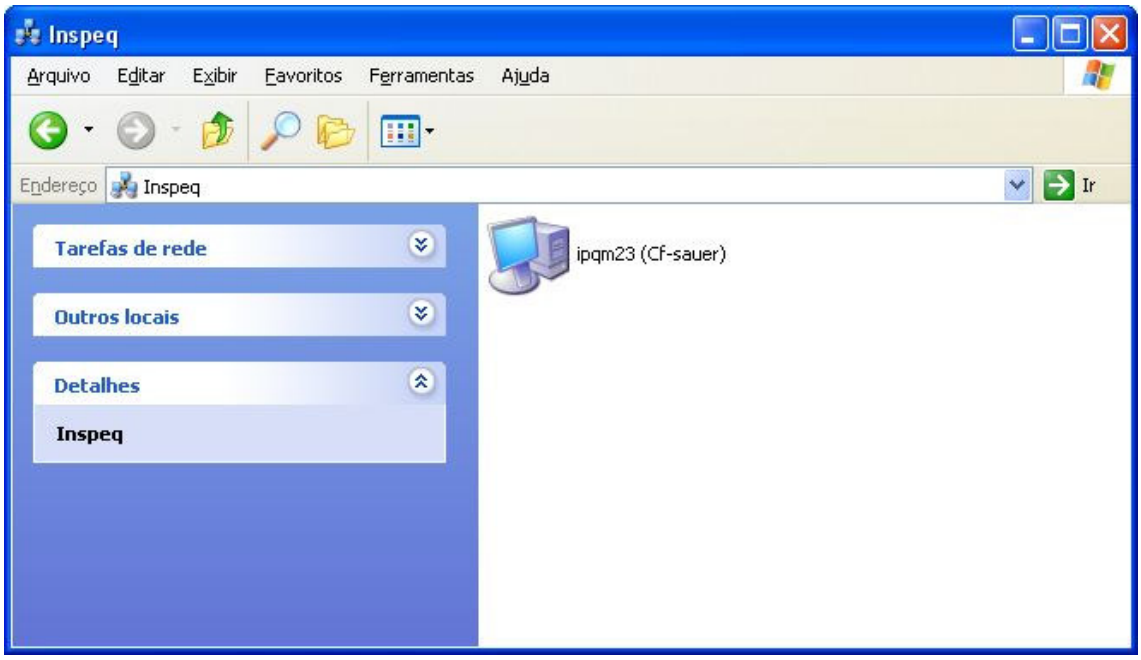


Figura 50 - Usuário não vê estações, mas acessa alguns serviços via *gateway*

A principal reclamação será intermitência, mais facilmente visível para o usuário do que a lentidão. A inserção deste panorama na aplicação CBR gerou a indicação de erro na máscara. As figuras 51 e 52 ilustram, respectivamente, a recuperação do caso no CBR e o diagnóstico detalhado, visando a solução do problema.

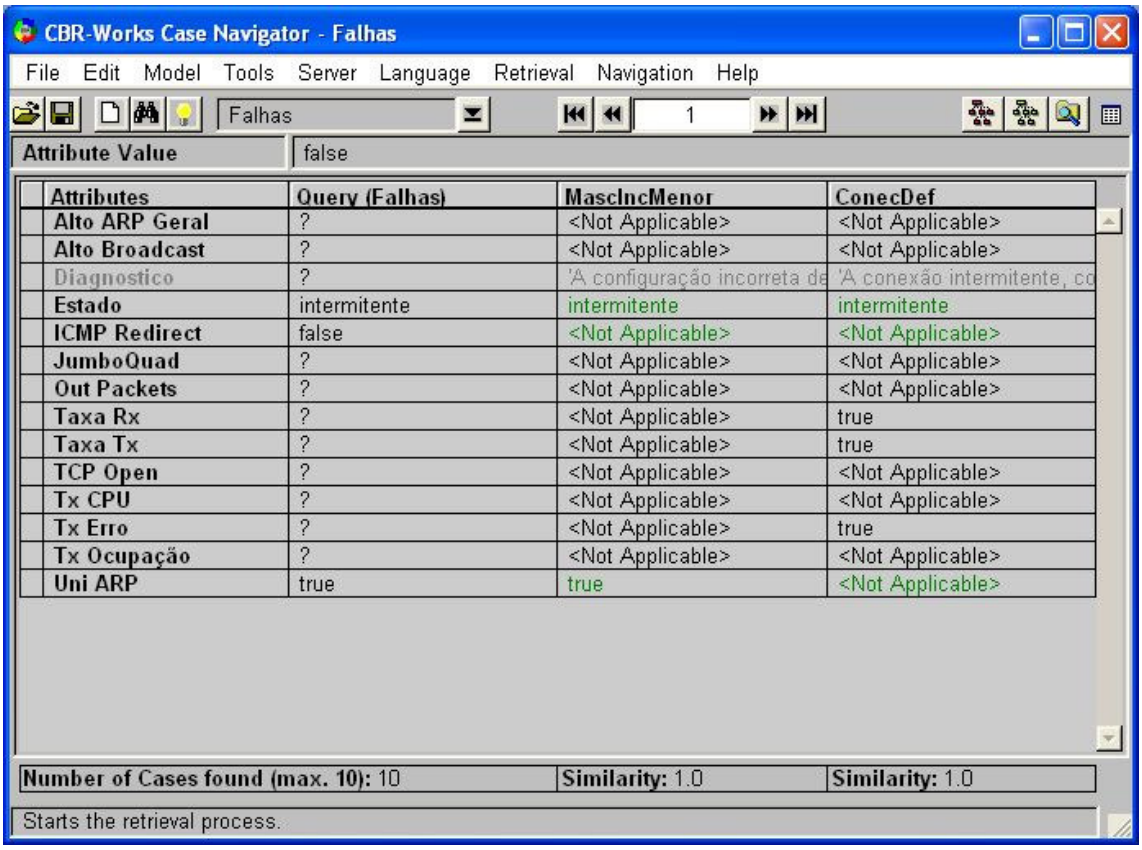


Figura 51 - O alto ARP de uma única estação é sinal relevante

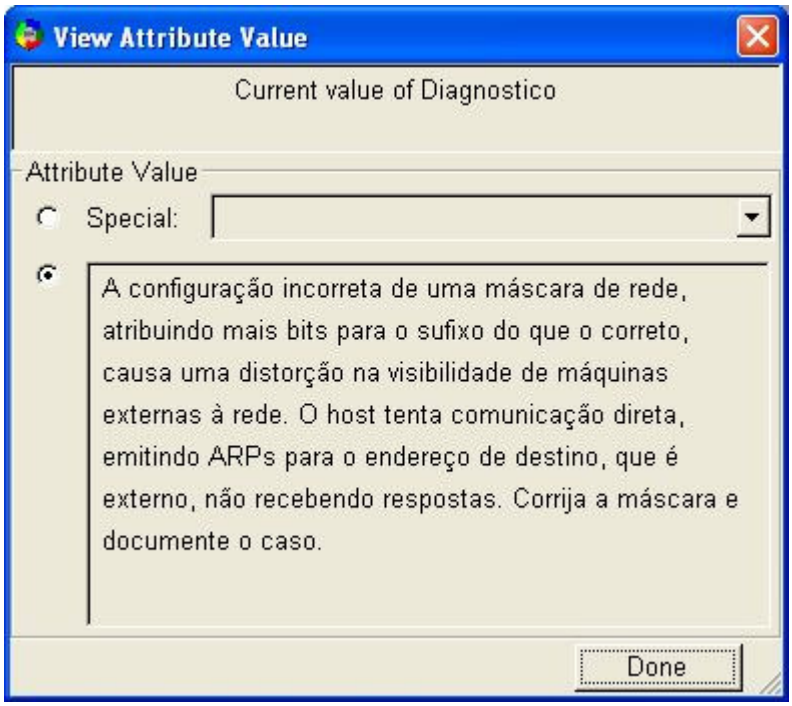


Figura 52 - O diagnóstico de Máscara Menor

O segundo cenário típico é o do erro na máscara para "maior", ou seja, aumentando o número de bits do prefixo. Tal erro ocasiona uma visibilidade limitada da rede, fazendo a estação interprete endereços como o 10.7.47.234, originalmente pertencente à mesma rede que a máquina 10.7.32.23, serão buscados via *Default Gateway*. Para exercitar esta situação, foi instalado um *ftp server* na máquina 10.7.47.234. A figura 53 ilustra o funcionamento correto deste servidor. Outro aspecto relevante, na mesma figura, é a inclusão na tabela ARP de todos os endereços de enlace das máquinas da rede com as quais a estação se comunicou.

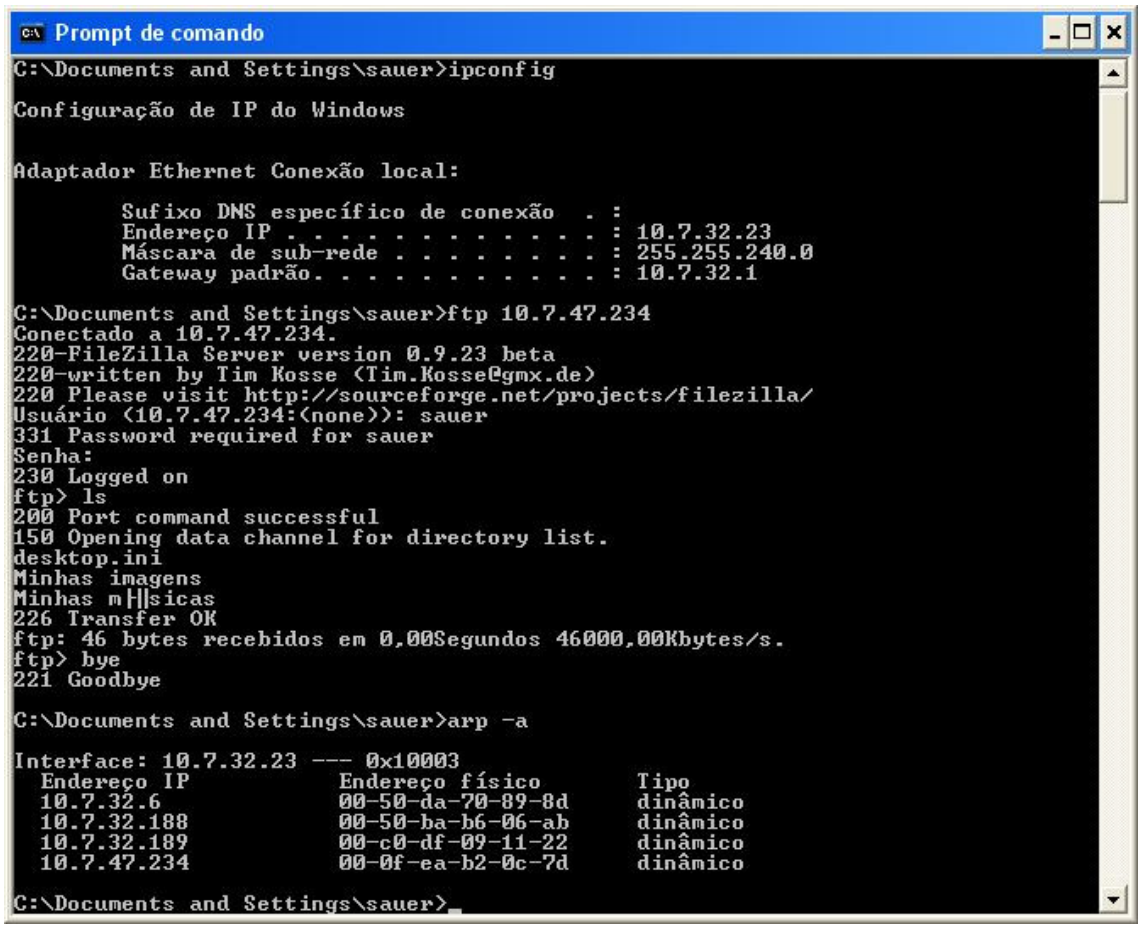


Figura 53 - Acesso a um *FTP Server* antes da alteração da máscara para maior

A seguir, foi alterada a máscara de subrede com um erro típico de digitação, trocando o 240 pelo 255 no terceiro *byte* da máscara. A figura 54 ilustra esta alteração.

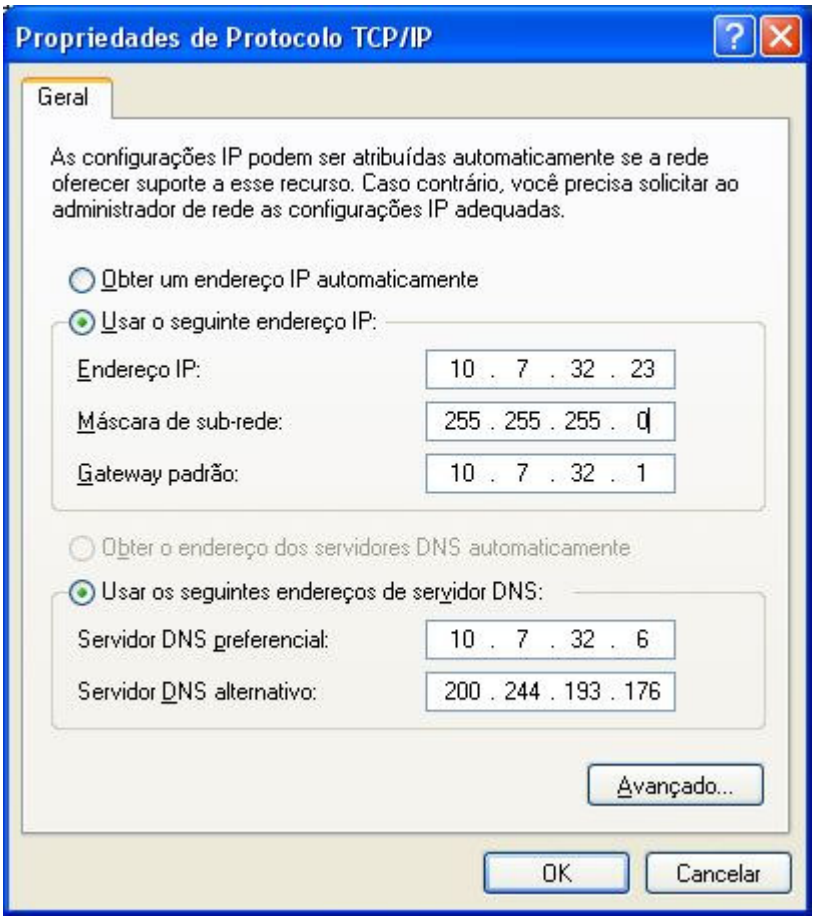


Figura 54 - Máscara maior, erro mais freqüente

Com isso, a estação entra em estado de intermitência, porque, mais uma vez, os endereços dos serviços mais comuns da rede, iniciados por 10.7.32, permanecem normalmente acessíveis. No entanto, a tentativa de acessar o servidor *ftp* instalado na máquina 10.7.47.255 gera um datagrama *ICMP redirect*, não apenas sobrecarregando a rede, como também o *default gateway*, que apenas deveria ser usado para acesso a endereços externos. Isso pode ser observado na figura 55, que ilustra a captura do tráfego com a evidenciação do tráfego ICMP indesejado. A figura 56 permite a compreensão da sobrecarga do *default gateway*. Apenas são guardados na tabela ARP os endereços dentro do escopo local, sendo então necessária a intervenção do *gateway* sempre que um endereço acima de 10.7.32.255 (que é um endereço de máquina, e não o *broadcast* da rede).

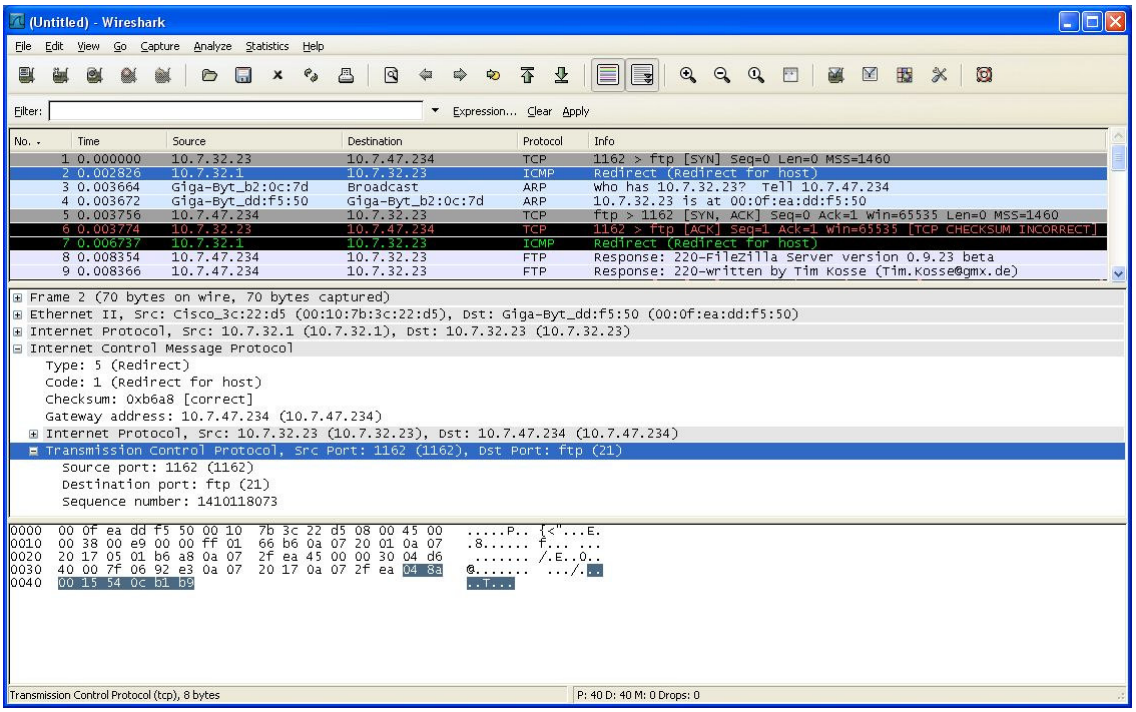


Figura 55 - Ocorrência de ICMP *Redirect* para endereços pertencentes à Rede

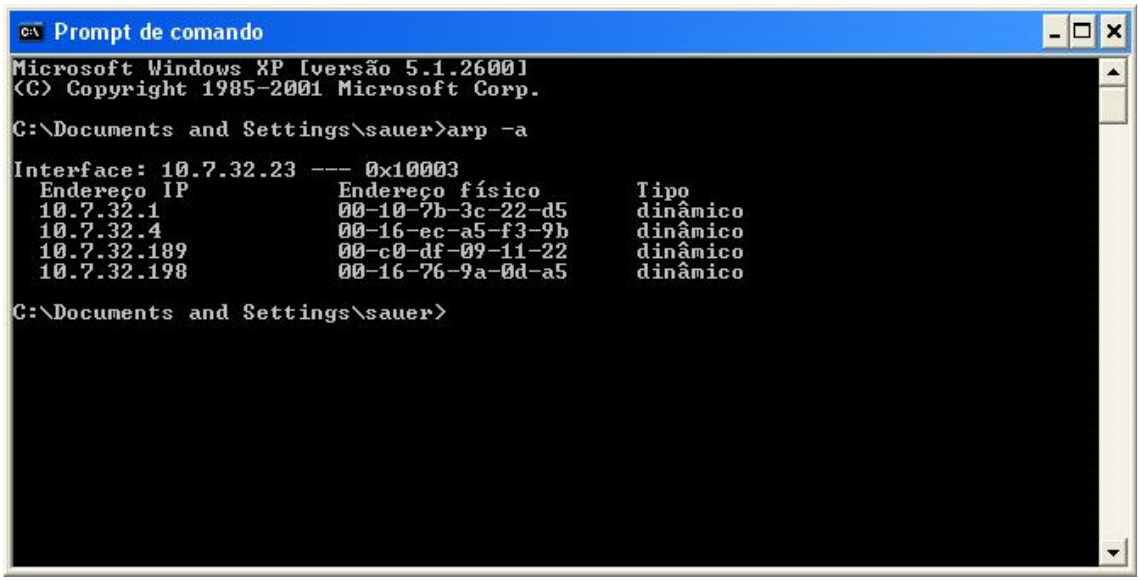


Figura 56 - Estação com Máscara maior não grava ARP de alguns endereços internos

A submissão das características deste caso ao CBR, mais uma vez, conduziu ao diagnóstico esperado, permitindo assim uma rápida recuperação da falha por qualquer técnico de suporte, mesmo que não tivesse passado por situação similar. A figura 57 apresenta os resultados da recuperação do caso mais similar na aplicação CBR desenvolvida neste trabalho. A figura 58 descreve precisamente o diagnóstico e a solução mais indicada. Mesmo que esta solução não resolvesse o problema, poder-se-ia tentar o próximo caso, até resolvê-lo definitivamente.

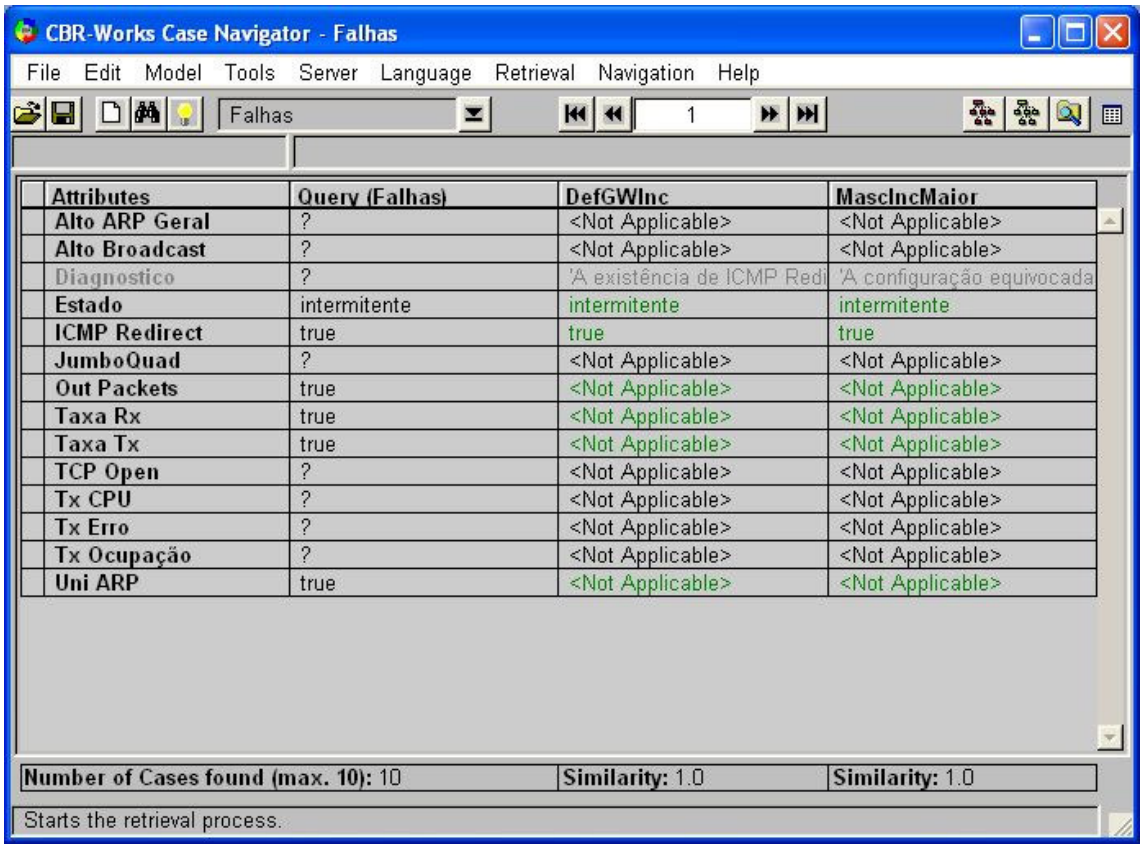


Figura 57 - Busca do Caso mais similar

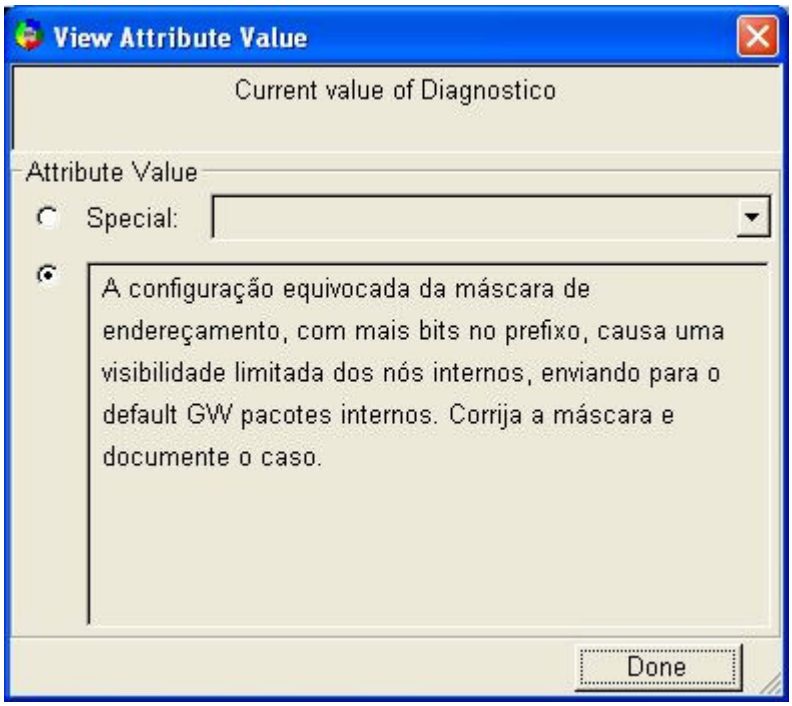


Figura 58 - Diagnóstico pelo CBR-Works

6.3 Caso 3 – Cabo com Defeito

A conectorização de *patch-cords*, aparentemente simples e meramente mecânica, depende de algumas precauções básicas para garantir sua funcionalidade adequada. Material de baixa qualidade, tanto nos seus componentes (cabo e conector) quanto na ferramenta (alicate de crimpagem) e mesmo a imperícia do responsável pela crimpagem pode provocar um problema de difícil solução. No momento da crimpagem, os cabos são tipicamente submetidos apenas a um teste de continuidade, e raramente há scanners de rede disponíveis para testes de NEXT (*Near-End Crosstalk*), impedância característica e existência de curtos. As aplicações mais usadas para gerenciamento de tráfego, como os *sniffers*, não contabilizam erros, uma vez que os mesmos são descartados na unidade de enlace. Para descobri-los, é necessário acessar os comutadores para obter as estatísticas, ou então usar aplicações SNMP, como o MRTG (OETIKER, 2006) que possibilitem o acesso às MIBs (DECKER *et al.,* 2006) que acumulam erros de CRC, como por exemplo, através da OID (*Object ID*) *ifInErrors*, 1.3.6.1.2.1.2.2.1.14. Em um cenário que representa a maioria das instalações, a equipe de suporte é sobrecarregada e não consegue agir de forma pró-ativa, monitorando tais informações para correção de erros antes da manifestação do usuário.

Figura 59 - Cabo irregular

Para testar a situação descrita, foi propositadamente preparado um cabo com os fios de sua extremidade com aproximadamente 50mm destrançados, visando aumentar reduzir o valor do parâmetro NEXT, que por si só já provocará distorção nos sinais. Além disso, alguns fios foram descascados para possibilitar a criação artificial de curtos, que ocorreriam, por exemplo, em um conector crimpado sem a força adequada ou com alicates desregulados. A figura 59 ilustra o conector usado para o teste.

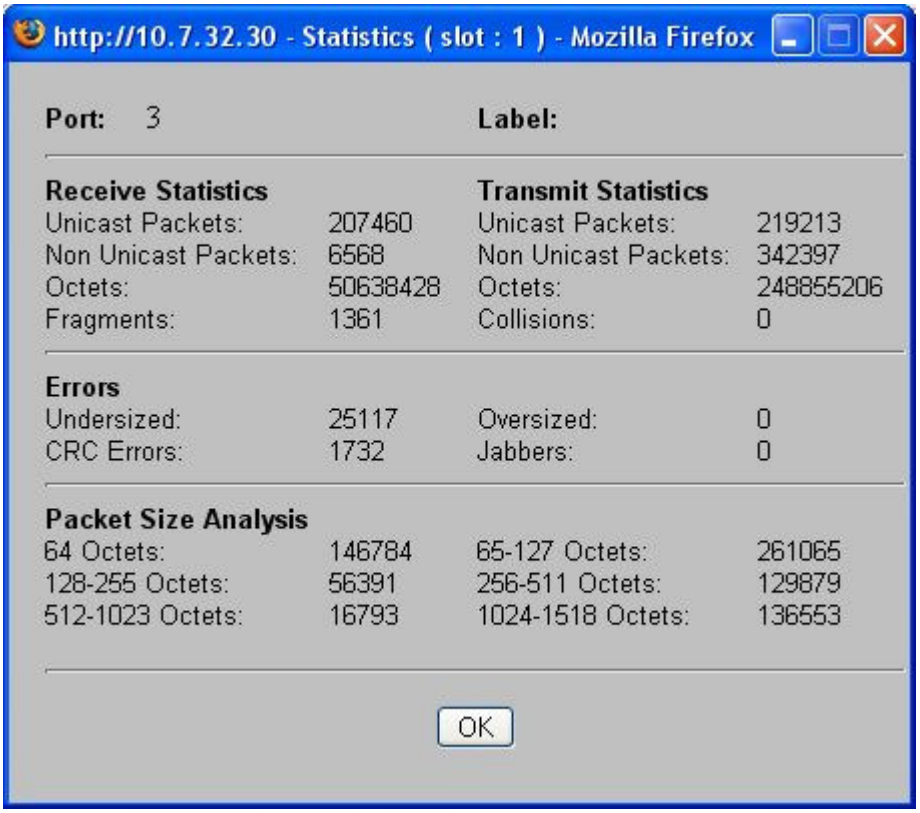


Figura 60 - Antes do teste

Para aferir a condição de erro, inicialmente foi capturada uma tela da aplicação de administração do comutador onde o *patch cord* foi instalado. Os erros que aparecem são decorrentes de uma operação ininterrupta desta rede por mais de 180 (cento e oitenta dias), com requisitos de alta disponibilidade, bem como da inserção de erros durante e imediatamente após a substituição de um cabo correto pelo cabo de teste. Esta tela é apresentada na figura 60. Após 2 (dois) minutos de teste, e algumas reclamações de usuários reportando lentidão e intermitência na rede, foi capturada a tela ilustrada na figura 61. Para 61437 pacotes recebidos nesta interface, 2879 erros ocorreram, representando aproximadamente 4,7 % do tráfego. Conforme discutido no capítulo 5, em uma rede saudável a taxa de erros deve estar muito próxima de zero.

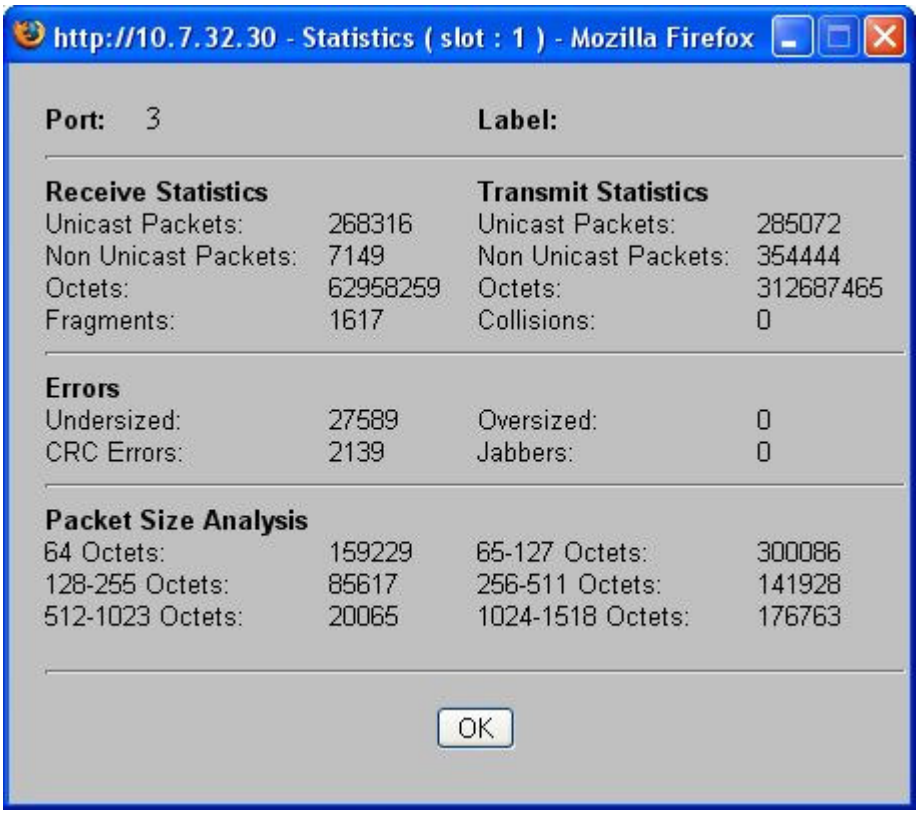


Figura 61 - Depois de 2 min de teste

Com a situação simulada, foram usados os sintomas percebidos para obtenção dos casos similares na aplicação CBR, obtendo-se os resultados ilustrados nas figuras 62 e 63.

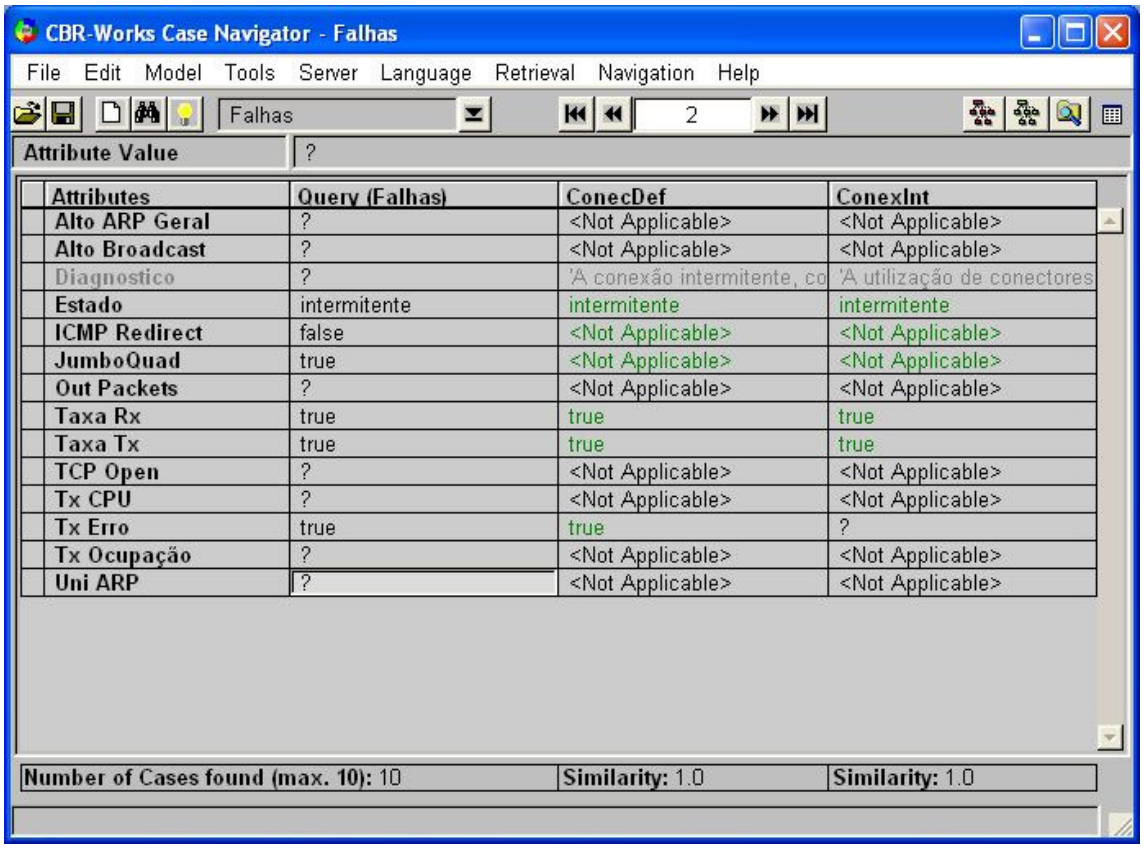


Figura 62 - Busca no CBR

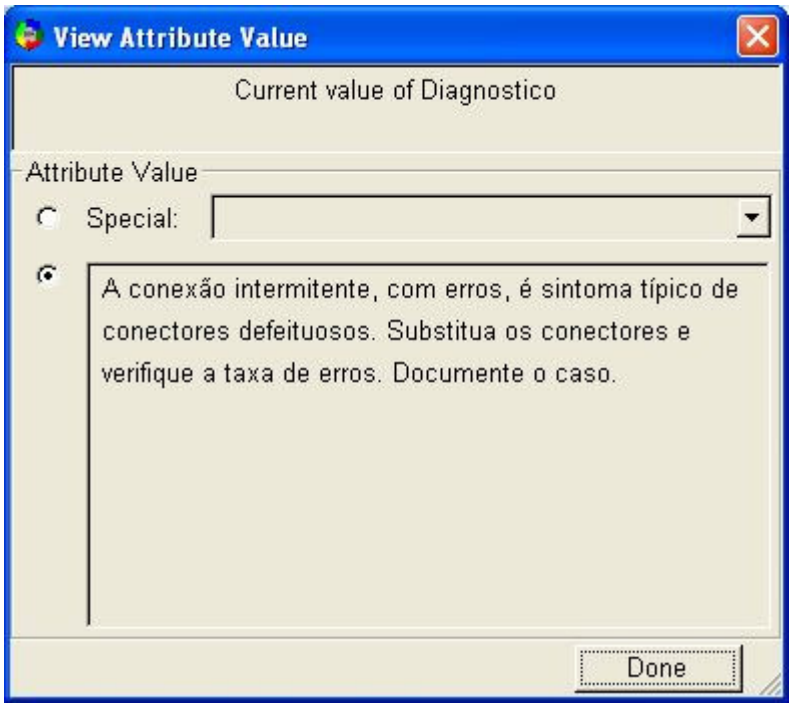


Figura 63 - Diagnóstico para cabo defeituoso

6.4 Resultados Obtidos

Após o encerramento da aquisição de conhecimento inicial, através dos casos típicos descritos no apêndice, bem como nos casos já resolvidos e armazenados no sistema legado, O sistema foi então levado para um segundo nível, visando não apenas o teste de sua viabilidade como também o treinamento do pessoal. Foi estabelecido um procedimento formal de registro dos chamados, através de um único contato para a solução de todos os problemas na rede local. A papeleta ilustrada na figura 64 permitiu não apenas o registro das situações de falha, mas também uma maior organização do setor, e até a avaliação do nível de compreensão do sistema CBR pelos técnicos.

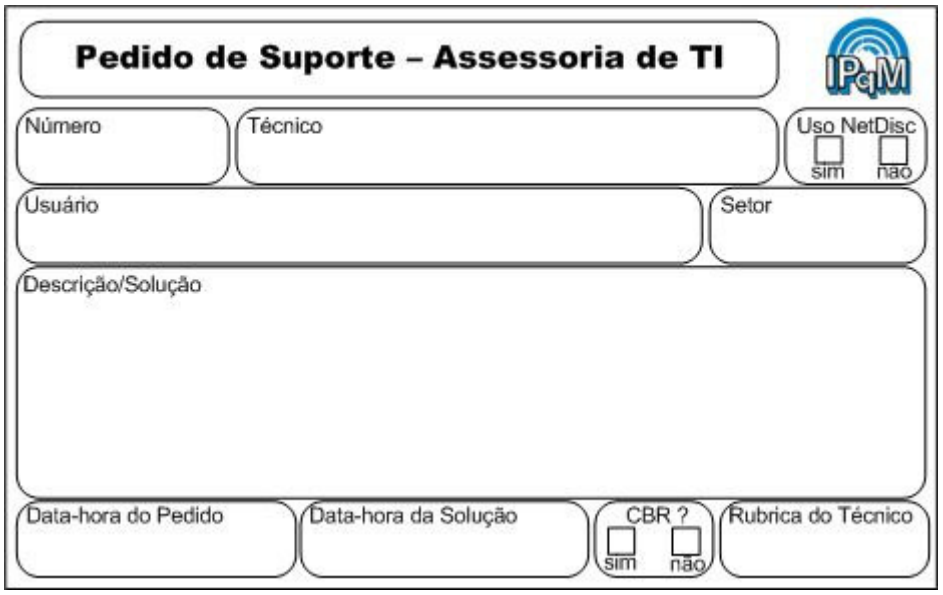


Figura 64 - Papeleta para Registro de Pedido de Suporte

Na papeleta da figura 64, além das principais informações do chamado, a eventual utilização da aplicação para descoberta de informações da rede (NetDisc – *Network Information Discovery*) é marcada, de forma a possibilitar a aferição da utilidade da ferramenta. Caso não se trate de uma ocorrência de falha, e sim de um suporte trivial não relacionado com os objetivos do trabalho, marca-se a opção "não", na caixa "CBR ?".

Durante seis semanas o sistema foi utilizado da forma estabelecida, atendendo aos chamados que foram considerados pertinentes, ou seja, aqueles que efetivamente se tratavam de falhas, e não, por exemplo, solicitações de novas configurações, instalações de software ou treinamento. Para possibilitar uma avaliação de eficiência do sistema, os técnicos foram orientados a resolver inicialmente o problema da forma convencional, com o registro do tempo decorrido para solução. Após isso, era realizado o procedimento através do Sistema CBR, fazendo-se as tentativas de solução do problema na seqüência indicada de acordo com o índice dos casos similares. Em todos os casos, o usuário era estimulado a utilizar a ferramenta de descoberta de informações da rede, para possibilitar uma descrição mais rica, o que naturalmente facilitaria a busca do caso mais similar. Independentemente do resultado, o caso é inserido na base, para seu enriquecimento com conhecimento especialista.

Durante o período de teste, foram recebidas 83 (oitenta e três) chamadas de uma significativa amostra das 400 estações desta rede, todo o setor da administração, aproximadamente metade das estações, utilizadas por usuários cuja *expertise* na solução de problemas se assemelha a dos usuários típicos de qualquer empresa. Destas

requisições de suporte, 51 (cinqüenta e um) casos foram pedidos de criação de contas e questões de direitos de acesso, reparos e *upgrades* de hardware de estações de trabalho, pedidos de criação de novos pontos de rede, dentre outros menos freqüentes. Os 32 (trinta e dois) casos restantes possuíam a natureza focada nos objetivos deste trabalho. Estes casos se tratavam de ocorrências de falhas na rede, principalmente devido a sua obsolescência, falta de investimento em equipamentos de certificação e deficiência numérica de pessoal para a adoção de uma postura pró-ativa no tratamento destas falhas. Dentre estes 32 casos, utilizando-se a metodologia de avaliação proposta, obteve-se o resultado ilustrado na tabela 6.

Tabela 6 - Resumo dos Casos Relevantes Registrados de 09/04/07 a 18/05/07

Data-hora início	***Data-hora diagnóstico***	***Sintomas***	***NetDisc***	***Sinais/testes***	***Diagnóstico***
100915P/ABR	100925P/ABR	Rede Fora	Sim	Acessa *web* e *intranet* normalmente e NetDisc evidencia erro	Máscara errada (maior) - R
101000P/ABR	101020P/ABR	Rede Lenta	Não	*Leds* piscando incessantemente	*Switch* sofrendo tempestade de *broadcasts* - E
111300P/ABR	111315P/ABR	Rede Lenta	Sim	Tudo ok, *Leds* piscando incessante-mente nos *switches* e NetDisc ok	ARP *unicast* causado por endereço errado (fora do escopo) - R
130930P/ABR	130955P/ABR	Rede Fora	Sim	Cabo ok, mas *leds* não acendem e NetDisc não vê info	Placa queimada - F
131410P/ABR	131420P/ABR	Rede Intermitente	Sim	Mexendo no cabo os *leds* acendem e apagam e NetDisc ok	Conector defeituoso - F
131500P/ABR	131545P/ABR	Rede Fora	Não	*Led* de continuidade aceso mas sem *flashing* de tx/rx	*Driver* corrompido - E
161400P/ABR	161420P/ABR	Rede Intermitente	Não	Mexendo no cabo os *leds* acendem e apagam	Conector defeituoso - E
171005P/ABR	171015P/ABR	Rede Intermitente	Sim	Aplicações com erro de *time-out* e NetDisc ok. Mexendo no cabo os *leds* acendem e apagam	Cabo mal-conectorizado, com intermitências de contato - F
171340P/ABR	171520P/ABR	Rede Fora	Não	Todo um segmento fora	*Transceiver* queimado - F
180800P/ABR	180850P/ABR	Rede Fora	Sim	Nenhum erro perceptível e Netdisc ok	Erro na digitação da senha - A
190750P/ABR	190825P/ABR	Rede Lenta	Sim	Incremento na taxa de erros no *switch* e NetDisc não responde	Placa de rede com defeito - F
190945P/ABR	190955P/ABR	Rede Lenta	Sim	*Leds* piscando incessantemente e NetDisc ok	ARP *unicast* causado por endereço errado (fora do

					escopo) - R
201100P/ABR	201125P/ABR	Rede Lenta	Não	Tudo físico ok, outros usuários no segmento acessando normalmente	Problema no URL de destino (Receita Federal) - A
251110P/ABR	251120P/ABR	Rede Intermitente	Sim	Mexendo no cabo os *leds* acendem e apagam e NetDisc ok	Conector defeituoso - F
251300P/ABR	251345P/ABR	Rede Lenta	Sim	Nenhum problema físico visível, taxa de erros incremental e NetDisc ok	Rota do cabo inadequada - F
251315P/ABR	251325P/ABR	Rede Lenta	Sim	Tudo físico e lógico ok, outros usuários no segmento acessando normalmente e NetDisc ok	Problema no acesso ao site de destino (SERPRO no fim do mês) - A
260745P/ABR	260750P/ABR	Rede Fora	Sim	Todo segmento fora. NetDisc não responde	*Switch* desligado - F
261025P/ABR	261100P/ABR	Rede Intermitente	Sim	Tudo físico e lógico ok, outros usuários no segmento acessando normalmente e NetDisc ok	Configuração do *browser* sem *proxy* - A
270745P/ABR	270820P/ABR	Rede Fora	Sim	Todo um segmento fora. NetDisc não responde	Porta de *switch* queimada - F
020920P/MAI	020945P/MAI	Rede Fora	Sim	Tudo ok, máquinas próximas ok, NetDisc ok.	Configuração do cliente de rede incorreta - A
021010P/MAI	021015P/MAI	Rede Intermitente	Sim	Aplicações com erro de *time-out*, NetDisc ok.	Cabo mal-conectorizado, com intermitências de contato - F
081120P/MAI	081140P/MAI	Rede Intermitente	Sim	Aplicações com erro de *time-out*, NetDisc ok.	Segmento congestionado por *broadcasts* - E
090750P/MAI	090830P/MAI	Rede Lenta	Sim	Nenhum problema físico visível, taxa de erros incremental, NetDisc ok.	Rota do cabo inadequada - F
101550P/MAI	101555P/MAI	Rede Intermitente	Sim	Acesso interno normal, sem acesso externo, Netdisc evidencia erro.	DNS digitado errado - R
110825P/MAI	110840P/MAI	Rede Lenta	Sim	Tráfego *unicast* ARP. NetDisc ok.	Contaminação por *worm* - A

110830P/MAI	110840P/MAI	Rede Fora	Sim	Todo um segmento fora. NetDisc não responde.	*Switch* desligado - F
111005P/MAI	111025P/MAI	Rede Lenta	Sim	Tudo físico aparentemente ok, lógico ok, alta taxa de erros. NetDisc ok.	Porta de *switch* com defeito intermitente - F
141350P/MAI	141430P/MAI	Rede Lenta	Sim	Nenhum problema físico visível, taxa de erros incremental, NetDisc ok.	Rota do cabo inadequada - F
160915P/MAI	160925P/MAI	Rede Lenta	Não	Tráfego *multicast* desconhecido na rede	Configuração de roteador errada - R
180740P/MAI	180755P/MAI	Rede Intermitente	Sim	Aplicações com erro de *time-out*, NetDisc ok.	Cabo mal-conectorizado, com intermitências de contato - F
181330P/MAI	181340P/MAI	Rede Lenta	Sim	Erros no TCP (CRC), NetDisc ok.	Placa da estação com defeito - F
181510P/MAI	181525P/MAI	Rede Intermitente	Sim	Mexendo no cabo os *leds* acendem e apagam, NetDisc ok.	Conector defeituoso - F

Computando-se dados da tabela 6, é possível sumarizar alguns resultados relevantes, demonstrados na tabela 7.

Tabela 7- Sumário dos Resultados do uso do CBR e NetDisc

Casos Físicos	17 (53 %)
Casos de Enlace	04 (12,5 %)
Casos de Rede	05 (15,7 %)
Casos de Aplicação	06 (18,8 %)
Tempo médio de recuperação problemas Físicos	22 min
Tempo médio de recuperação problemas Enlace	26 min
Tempo médio de recuperação problemas Rede	10 min *
Tempo médio de recuperação problemas Aplicação	26 min
Tempo médio de recuperação	21 min
Tempo médio com uso do NetDisc	20 min
Tempo médio sem o uso do NetDisc	27 min

* - Todos resolvidos através do diagnóstico com o NetDisc, de forma trivial.
Tempo médio típico de resolução de falhas: 45 minutos
Ganho médio em performance: 24 minutos (53,3 %)

Na tabela 7, é possível observar as seguintes características da abordagem adotada:

- A eficiência da equipe aumentou em mais de 50%;
- O uso do NetDisc permitiu diagnósticos rápidos, que na maioria das vezes foram realizados remotamente;
- O diagnóstico via NetDisc para problemas de rede foi, na maioria das vezes, imediato (média 10 minutos); e
- Comprovando-se a expectativa descrita durante o levantamento bibliográfico, mais da metade dos problemas foram físicos.

Em função destes resultados, pode-se emitir o seguinte juízo de valor:

- O uso da ferramenta NetDisc é viável, e o investimento em seu aprimoramento pode reduzir ainda mais o tempo de indisponibilidade das estações de trabalho;
- O Raciocínio Baseado em Casos é uma excelente ferramenta para Gerenciamento de Redes. Em função dos sintomas, sinais podiam ser imediatamente buscados, não mais de forma aleatória, mas sim agora de forma sistemática e com um fim definido. A figura 65 ilustra a possibilidade de se completar um caso através de perguntas em busca de sinais. Estes testes foram introduzidos no CBR, conforme a política de aumento da base de conhecimento adotada; e

- A significativa redução do tempo de indisponibilidade, com uma equipe ainda pouco treinada e naturalmente resistente às mudanças, é encorajadora. Acredita-se que esta performance ainda melhorará mais, com a incorporação definitiva da ferramenta para análise do tráfego ICMP e a adaptação da equipe à aplicação CBR.

Uma comparação relevante, que confirma o valor agregador deste trabalho pode ser feita com o Sistema Homer (BERGMANN, GÖKER, 2003). Através do Homer, foi obtida uma eficácia de 32% das submissões, em um universo de 57 casos, ou seja, na mesma ordem de grandeza que a aplicação deste trabalho, onde a eficácia atingiu 100%.

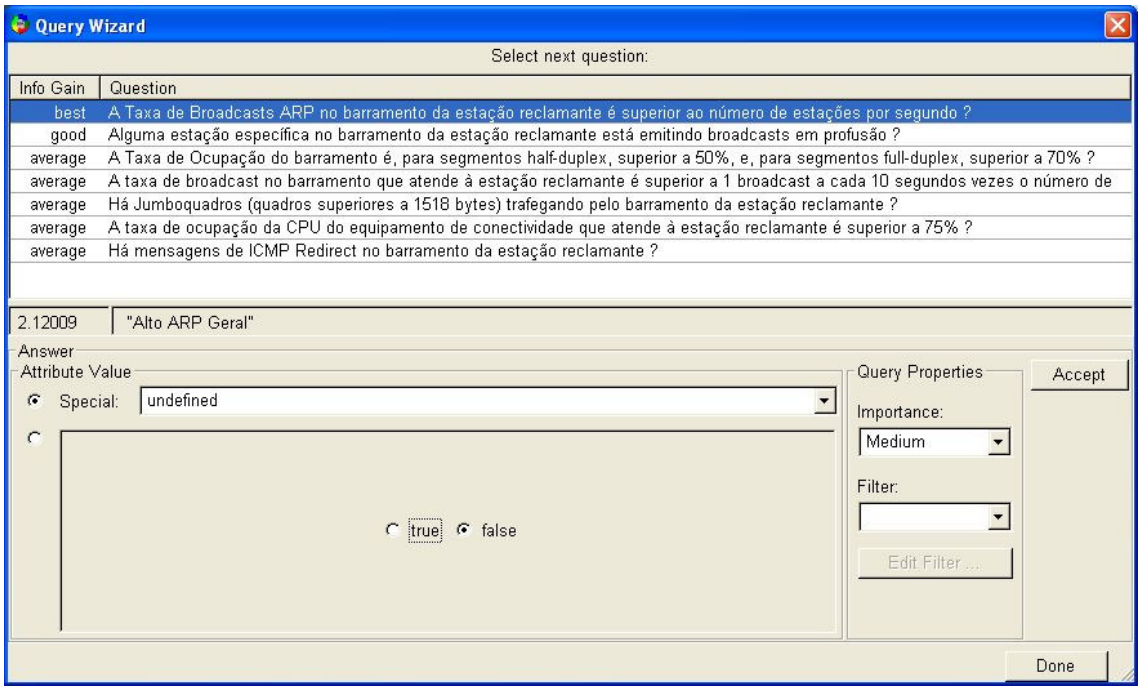


Figura 65- Busca de Sinais através de Testes Confirmatórios

A aplicação para tratamento de ICMP, no seu primeiro protótipo, apresentou pouco benefício em relação às suas conseqüências. A colocação da interface em modo promíscuo torna o ambiente do usuário mais carregado. Além disso, a necessidade da filtragem e análise contínua dos pacotes ICMP pela estação agravou a sensação de lentidão, chegando até a ser notificada como falha por alguns usuários. Foi então implementada uma nova versão experimental, não documentada neste trabalho, porém já disponível e em testes. Neste novo protótipo os pacotes ICMP são apenas verificados quanto ao seu tipo, e aqueles mais típicos de indicativos de falhas são enviados via FTP para um servidor onde uma aplicação processa os pacotes de todas as estações. A figura 66 ilustra esta segunda versão-protótipo.

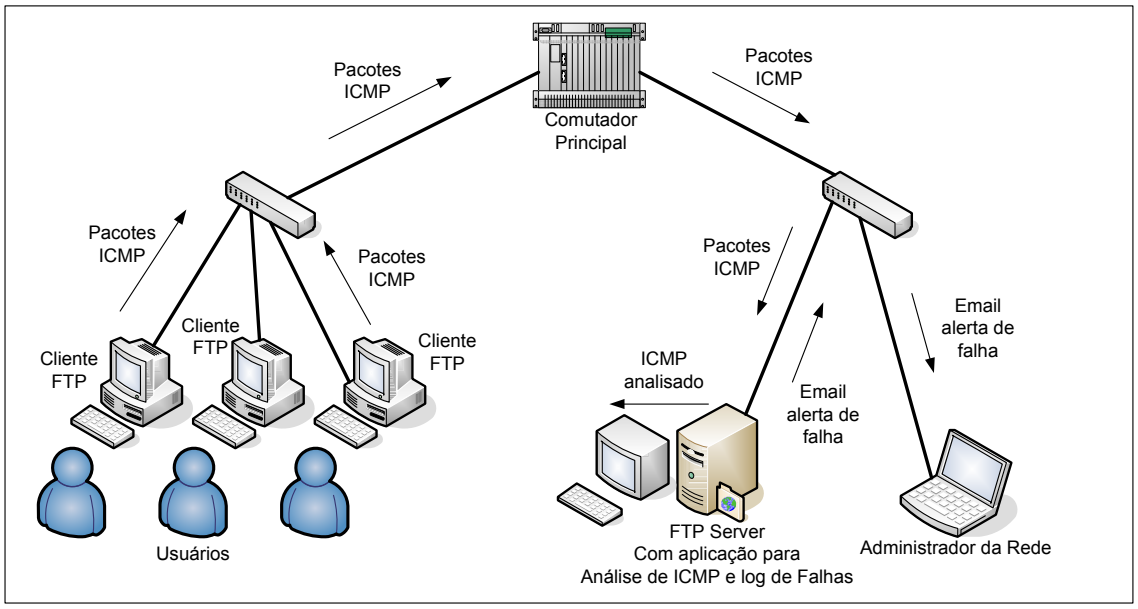


Figura 66 - Segunda versão do Analisador de ICMP

Como se pode observar na figura 66, as ocorrências de ICMP são enviadas pelas estações de trabalho, através dos seus respectivos clientes FTP, para um servidor FTP único para toda a rede. Neste equipamento, há uma aplicação para análise dos pacotes ICMP recebidos. Caso esta aplicação "servidor" detecte uma falha, emite um alerta via console, armazena a ocorrência em *log* e despacha um *email* para o administrador com a informação do sintoma percebido, bem como as informações da estação envolvida. Este protótipo já permitiu uma redução satisfatória na sobrecarga das estações de trabalho, porém ainda utiliza o procedimento de colocar a interface das estações de trabalho em modo promíscuo. Outro aspecto relevante é que há uma replicação de pacotes ICMP em direção ao servidor, contribuindo para congestionar a rede. Na época da redação desta conclusão, algumas possibilidades de solução para este problema já estavam em estudo, como por exemplo o tratamento mais simples dos pacotes ICMP nas estações, visando reduzir ainda mais a sobrecarga na rede.

6.5 Conclusões dos Estudos de Caso

Os casos ilustrados neste capítulo foram abrangentes, já que abordaram falhas dentro das camadas do modelo OSI onde os usuários tipicamente mais percebem os sintomas e se motivam a pedir apoio. Os resultados foram os esperados, já que a inserção dos sintomas e sinais na aplicação CBR permitiu a recuperação de casos similares, com indicação precisa do diagnóstico e da solução proposta.

O próximo capítulo apresentará as conclusões finais, baseadas nos resultados obtidos durante toda esta pesquisa de campo, ressaltando aspectos importantes para que sua contribuição possa ser útil para minimizar o tempo de indisponibilidade das redes e das estações de trabalho. Algumas observações sobre possíveis extensões do trabalho, incluindo sugestões de aprofundamento em alguns tópicos importantes, porém não abordados, serão definidos.

7 Conclusões e Trabalhos Futuros

O ambiente heterogêneo e não-determinístico do tráfego das redes de computadores torna desafiadora a tarefa de gerenciamento. Após a pesquisa bibliográfica, onde se concluiu que a estratégia fundamentada em Raciocínio baseado em Casos era interessante e podia incrementar a eficiência e a eficácia das equipes de suporte, optou-se pela implementação de um sistema CBR, cujos testes poderiam ser validados com o uso de dados reais. Para tanto, foi utilizado um ambiente complexo de um Instituto de Pesquisas, com aproximadamente 400 (quatrocentas) estações de trabalho, bem como a sua equipe de suporte. Para aperfeiçoar o sistema e agregar uma maior contribuição a este trabalho, foram idealizadas, projetadas e implementadas duas ferramentas auxiliares. A primeira ferramenta captura pacotes ICMP recebidos pelas estações de trabalho e os analisa, em busca de sintomas de falhas. A segunda realiza uma leitura de informações nem sempre óbvias para a maioria dos usuários, porém vitais para o diagnóstico de falhas em redes, como por exemplo os endereços configurados e a existência de conectividade com os elementos topológicos chave da rede. Os resultados foram encorajadores, uma vez que permitiram as seguintes agregações de valor objetivas:

- A experiência cotidiana dos técnicos de suporte passa a ser compartilhada pelos seus pares, através da recuperação de casos armazenados;
- Redução importante do tempo necessário para o diagnóstico e a solução de falhas, aumentando a produtividade da equipe de suporte e a qualidade do tráfego na rede; e
- Em função da documentação detalhada dos casos de forma supervisionada, a qualidade técnica dos atributos de cada falha é elevada, possibilitando aos técnicos um melhor entendimento das razões pelas quais determinada irregularidade na rede causa uma falha. Este aspecto em particular agregou vantagens decorrentes. O aprimoramento técnico dos elementos do setor de suporte possibilitou que os mesmos adotassem uma postura pró-ativa, buscando irregularidades antes que as mesmas viessem a se tornarem falhas. Isso pode ser observado pela redução dos pedidos de suporte atuais, mesmo

sem a ocorrência de investimentos e o contínuo aumento da demanda de banda pelas aplicações.

Como sugestões para trabalhos futuros, uma excelente funcionalidade dos sistemas CBR não explorada neste trabalho foi o ajuste da métrica de similaridade. Conforme descrito nos capítulos 3 e 5, as variações de modelos de cálculo de similaridade poderão alterar a indexação dos casos indicados como possíveis soluções, permitindo uma redução ainda maior no tempo de solução de falhas. A continuidade deste trabalho, com algum investimento no campo da similaridade, certamente possibilitará o alcance de resultados ainda melhores. As ferramentas auxiliares, apesar de protótipos, sem a pretensão de equivaler-se aos aplicativos comerciais, obtiveram sucesso em um bom número de casos, já que o objetivo era tão somente reduzir o tempo de diagnóstico e solução. O aprimoramento das mesmas, eventualmente através da incorporação ao ambiente de software básico das estações de trabalho, poderá obter melhorias nos resultados obtidos, especialmente no que diz respeito à sobrecarga tanto do processamento das máquinas quanto da banda disponível da rede.

O *Shell* CBR (WATSON, 1997) também possui toda uma sorte de funcionalidades não exploradas neste trabalho. Após atingir um nível satisfatório de maturidade, a aplicação desenvolvida pode ser disponibilizada na rede através de uma interface *web*, ampliando ainda mais o escopo de sua utilidade.

Em um cenário globalizado, as Redes de Computadores são imprescindíveis para a realização das tarefas mais banais do cotidiano dos cidadãos. Com isso, o Gerenciamento das Redes, principalmente através do rápido e correto diagnóstico e solução de falhas, passa a ser de grande importância. Com a qualidade necessária, as Redes podem contribuir para a inclusão digital, a democratização do conhecimento, a acessibilidade, e tantas outras iniciativas sócio-educativas. A ciência e a produção acadêmica, por exemplo, são beneficiárias diretas das Redes. A ciência, para a aceitação de novos conceitos, prescinde de divulgação, aceitação e uso por parte da comunidade acadêmica. Dispondo de Redes de Comunicação de Dados eficientes, toda sociedade se beneficiará.

Referências Bibliográficas

AGHARSARYAN, A., FABRE, E., BENVENISTE, A. et al., 1997, "A Petri Net Approach to Fault Detection and Diagnosis in Distributed Systems". In: *Proceedings of the 36th IEEE Conference on Decision and Control*, pp. 720-725, San Diego, USA, December.

ALTHOFF, K-D., MÄNS, J., NICK, M., 2005, "Maintaining Experience to Learn: Case Studies on Case-Based Reasoning and Experience Factory". In: *Proceedings of the Learning, Knowledge Discovery and Adaptativity Workshop*, pp. 118-125, Saarbrücken, Germany, October.

ANÔNIMO, 2001, ANSI/TIA/EIA-568-B series – "Commercial Building Wiring Standards and Addendum". Telecommunications Industry Association (TIA).

ANÔNIMO, "Ethereal – The world's most popular network protocol analyzer", disponível em <http://www.ethereal.com>. Acesso em 10 dez 2006.

ANONIMO, "US-CERT – United States Computer Emergency Readiness Team – Vírus Basic", disponível em http://www.us-cert.gov/reading_room/virus.html. Acesso em 05 mar 2007.

BAKER, F. "RFC 1812 – Requirements for IP Version 4 Routers", disponível em <http://rfc.net/rfc1812.html>. Acesso em 27 fev 2007.

BARAS, J.S., BALL, M., GUPTA, S., et al., 1997, "Automated Network Fault Management", In: *Proceedings of the MILCOM'97*, vol. 3, pp. 1244-1250, Monterey, November.

BASILI, V. R., CALDIERA, G., ROMBACH, H. D., 1994, "Experience Factory". In: *Encyclopedia of Software Engineering*, vol. 1, Ed. John Wiley & Sons, pp. 528-532.

BERGMANN, R., 2001,"Highlights of the European INRECA Projects". In: *Lecture Notes in Computer Science*, v.2080, Springer-Verlag, pp. 1-15.

BERGMANN, R., GÖKER, M., 2003, "Developing Industrial Case-Based Reasoning Applications with the INRECA Methodology". In: *Lecture Notes in Artificial Intelligence*, 2nd ed., v.1612, Part II, Springer-Verlag.

BERGMANN, R., ALTHOFF, K-D., 1998, "Methodology for Building CBR Applications", In: *Lecture Notes in Computer Science*, v. 1400, Springer-Verlag, pp. 299-326.

BERGMANN, R., BREEN, S., GÖKER, M. et al.,1998, "The INRECA-II Methodology for Building and Maintaining CBR Applications". In: *Proceedings of the 6th German Workshop on Case-Based Reasoning (GWCBR'98)*, Berlin, Germany.

BRODIE, M., RISH, I., MA, S., 2002, "Intelligent Probing: A Cost-effective approach to Fault Diagnosis in Computer Networks". In: *IBM System Journal*, Vol 41, No 3, pp. 372-385.

BRODIE, M., RISH, I., MA, S., 2002, "Accuracy vs. Efficiency Trade-offs in Probabilistic Diagnosis". In: *Proceedings of 18th National conference on Artificial Intelligence AAAI-2002*, pp. 560-566, Alberta, Canadá, July.

CASE, J., FEDOR, M., SCHOFFSTALL, M. et al. “RFC 1157 – Simple Network Management Protocol (SNMP)”, disponível em: <http://www.ietf.org/rfc/rfc1157.txt>. Acesso em 25 fev 2006.

CHAO, C. S., YANG, D. L.,.LIU, A.C., 1999, “An Automated Fault Diagnosis System Using Hierarchical Reasoning and Alarm Correlation”, In: *Proceedings of the IEEE Workshop on Internet Applications*, pp. 120-127, San Jose, USA, July.

CHAO, C. S., YANG, D. L.,LIU, A.C., 2001, “A LAN Fault Diagnosis System”, In: *Computer Communications*, v.24, No. 14, Elsevier, pp. 1439-1451.

CHAO, C. S., YANG, D. L., LIU, A. C., 2001, “A Time-aware Fault Diagnosis Systems in LAN”, In: *Proceedings of the International Symposium on Integrated Network Management*, pp. 499-512, Seattle, EUA, May.

CHAPPELL, L., FARKAS, D., 2003, Cisco Internetwork Troubleshooting,. 1st ed. Cisco Press.

CHEN, J-L., HUANG, P-H., 1996, “A Fuzzy Expert System for Network Fault Management”, In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 328-331, Beijing, China, October.

CHOUDHURY, P. P., 1999, “An Information Theoretic Model of a Reliable Computer Network”, In: *Applied Mathematics Letters*, v.12, Elsevier, pp. 125-130.

CHUTANI, S., NUSSBAUMER, H.J., 1995, “On the Distributed Fault Diagnosis of Computer Networks”. In: *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management, IV (ISINM'95)*, pp. 71-77, Egypt, June.

DALMI, I, KOVACS, I., LORANT, I., et al., 1998, “Adaptive Learning and Neural Network in Fault Diagnosis”, In: *Proceedings of the UKACC International Conference on Control*, Vol. 1, pp. 284-289, Swansea, September.

DECKER, E., LANGILLE, P., RIJSINGHANI, A. et al. “RFC 1493 – Definitions of Managed Objects for Bridges”, disponível em <http://www.ietf.org/rfc/rfc1493.txt>. Acesso em 26 fev 2006.

DUARTE, E. P., JR. DOS SANTOS, A. L., 2001,“Network Fault Management Based on SNMP Agent Groups”, In: *Proceedings of the International Conference on Distributed Computing Systems Workshop*. pp. 51-56, Mesa, USA, April.

FLICK, J., JOHNSON, J. “RFC 2665 – Definitions of Managed Objects for the Ethernet-like Interface Types”, disponível em <http://www.ietf.org/rfc/rfc2665.txt>. Acesso em 26 fev 2006.

GÖKER, M., ROTH-BERGHOFER, T., BERGMANN, et al., 1998, “The development of HOMER: A case-based CAD/CAM help-desk support tool”. In: *Advances in Case-Based Reasoning*, v. 1488, *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 346-357.

GOPAL, R., 2000, “Layered Model for Supporting Fault Isolation and Recovery”, *IEEE/IFIP. Network Operations and Management Symposium*, pp. 729-742, Honolulu, April.

HAUGDAHL, J. S., 2000, Network Analysis and Troubleshooting. 1st ed. Addison Wesley.

HAYES, C. CUNNINGHAM, P., DOYLE, M., 1998, "Distributed CBR using XML". *KI-98 Workshop on Intelligence Systems and Electronic Commerce*, No. LSA-98-03E, Bremen, Germany, September.

HUARD, J-F., 1993, *Research Proposal: Characterizing Network Complexity by Means of Fault Diagnosis*, Center for Telecommunications Research, Columbia University, Columbia, USA.

ISERMANN, R., BALLÉ, P., 1997, "Trends in the Application of Model-Based Fault Detection and Diagnosis of Technical Processes", In: *Control Engineering Practice*, v. 5, No. 5, Elsevier, pp. 709-719.

KUMAR, P.G., VENKATARAM, P., 1995, "Network Fault Diagnosis using a Realistic Abductive Reasoning Model", In: *Engineering Applications of Artificial Inteligence*, Vol. 28, No 6, Pergamon , pp. 703-708.

LAZAR, A. A., WANG, W., DENG, R.H., 1992, "Models and Algorithms for Network Fault Detection and Identification: A Review", *Singapore ICCS/ISITA*, vol. 3, pp. 999-1003, Singapore, November.

LECKIE, C., 1995, "Experience and Trends in AI for Network Monitoring and Diagnosis", In: *Proceedings of the IJCAI95 Workshop on AI in Distributed Information Networks*, Montreal, August.

LECKIE, C., SENJEN, R. WARD, B., et al., 1997,"Communication and Coordination for Intelligent Fault Diagnosis Agents", In: *Proceedings of the 8th IFIP/IEEE International Workshop on Distributed Systems Operation and Management*, pp. 280-291, Sydney, October.

LENZ, M., BURKHARD, H. D., 1996, "Case Retrieval Nets: Basic Ideas and Extensions". In: *Proceedings of the German Conference on KI-96: Advance in artificial intelligence*, Lecture Notes in Artificial Intelligence, Springer Verlag, pp. 227-239.

LENZ, M. BURKHARD, H. D., BRUECKNER, S., 1996, "Applying Case Retrieval Nets To Diagnostic Tasks in Technical Domains". In: *Advances in Case-Based Reasoning, Lecture Notes in Artificial Intelligence*, Springer Verlag, pp. 219-233.

LEWIS, L., 1993, "A Case-Based Reasoning Approach to the Management of Faults in Communications Networks", In: *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp.1422-1429, San Francisco, USA, March.

LOPES, R.V., SAUVÉ, J. P., NICOLLETTI, P. S., 2003, Melhores Práticas para Gerência de Redes de Computadores, 1ª ed. Rio de Janeiro, Ed. Campus.

MAEDA, C., 1992, "Categorization for Network Fault Diagnosis", In: *Proceedings of the 25th International Conference on System Sciences*, pp. 486-495, Kauai, January.

MCCLOGHRIE, K., KASTENHOLZ, F. "RFC 2233 – The Interfaces Group MIB using SMIv2", disponível em <http://www.ietf.org/rfc/rfc2233.txt>. Acesso em 26 fev 2006.

MELCHIORS, C., 1999, *Raciocínio Baseado em Casos Aplicado ao Gerenciamento de Falhas em Redes de Computadores*. Dissertação de M.Sc., PPGC da UFRGS, Porto Alegre, RS, Brasil.

OETIKER, T. "Multi Router Traffic Grapher", disponível em http://www.mrtg.org". Acesso em 26 fev 2006.

PEDRYCZ, W., CAMARGO, H., 2003, "Fuzzy Timed Petri Nets". In: *Fuzzy Sets and Systems*, v. 140, Issue 2, Elsevier, pp.301 – 330.

POSTEL, J. "RFC 792 – Internet Control Message Protocol", disponível em <http://www.ietf.org/rfc/rfc792.txt>. Acesso em 26 fev 2006.

POSTEL, J., MARKSON, T., SIMSON, B., et al. "ICMP Type Numbers". Disponível em <http://www.iana.org/assignments/icmp-parameters>. Acesso em 27 fev 2007.

ROMBACH. H. D., VERLAGE, M., 1995, "Directions in Software Process Research", In: *Advances in Computers*, v.41, Elsevier, pp. 1-63.

ROUVELLOU, I., HART, G. W., 1995, "Automatic Alarm Correlation for Fault Identification", In: *Proceedings of the 14th Annual Joint Conference of IEEE Computer and Communication Society*, pp. 553-561, Boston, EUA, April.

SABIN, M., RUSSELL, R. D., FREUDER, E. C., 1997, "Generating Diagnostic Tools for Network Fault Management", In: *Proceedings of the 5th IFIP/IEEE International Symposium on Integrated Network Management*, .pp. 700-711, San Diego, USA.

TANENBAUM, A. S. Computer Networks. 4th ed. Prentice Hall, 2003.

THOTTAN, M., CHUANYI, J., 2000, "Properties of Network Faults", In: *Proceedings of the IEEE/IFIP Network Operations and Management Symposium*, pp. 941-942, Honolulu, USA, April.

USHIO, T., ONISHI, I., OKUDA, K., 1998, "Fault Detection Based on Petri Net Models with Faulty Behaviors". In: *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cibernetics*, pp. 113-118, San Diego, USA, October.

VON WANGENHEIN, C. G., VON WANGENHEIN, 2003, A. *Raciocínio Baseado em Casos*. 1ª ed. São Paulo: Manole.

WATSON, I., 1997, CBR-Works In: *Applying Case-Based Reasoning*, cap. 6, Elsevier.

WILKE, W., BERGMANN, R., 1998, "Techniques and Knowledge Used for Adaptation during Case-Based Problem Solving". In: *Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Lecture Notes in Computer Science, Springer-Verlag, v.1416, pp. 497-506.

YOSHIDA, T., MOTODA, H., WASHIO, T., 2002, "Adaptative Ripple Down Rules Method based on minimum Description Length Principle". In: *Proceedings of the 2nd IEEE Data Mining (ICDM 2002)*, pp. 530-537, Maebashi City, Japan, December.

Apêndice 1 Detalhamento das Principais Falhas em Redes de Computadores

A-1.1 Problemas Físicos

Uma das falhas mais típicas de redes é a falta ou intermitência de continuidade entre condutores físicos de sinais. É também difícil de ser detectada visualmente pelo usuário, devido ao pequeno tamanho dos conectores e às proteções plásticas dos fios, que ocultam tais irregularidades. Os cabos de fibras óticas sofrem o mesmo problema, uma vez que podem ser flexionados além de um limite no qual ocorrem microfissuras que não provocam a total perda de continuidade, mas reduzem a qualidade do sinal causando perdas. Várias vezes o autor deste trabalho observou técnicos afirmarem a qualidade de um cabo defeituoso através de teste visual do feixe de luz, ineficaz para a avaliação da perda de dados em microfissuras. Por outro lado, sintomas de "rede lenta" foram solucionados através da pura e simples reconectorização do *patch-cord*.

A-1.2 Cabo interrompido

Descrição do Problema: Os cabos de rede, sejam eles de cobre ou fibra, são vulneráveis por estarem em algum momento expostos a ações mecânicas capazes de provocar danos. São famosos os casos de fibras óticas danificadas por roedores. Poucos técnicos responsáveis pela enfiação de cabos de par trançado sabem que os mesmos não podem ser submetidos à tração ou torcidos excessivamente, pois podem provocar alterações mecânicas que influenciam nos parâmetros de atenuação e *crosstalk* dos mesmos.

Sintomas típicos: A **falta de conectividade** é o sintoma típico principal. Em caso de microfissuras em fibras, cabos com continuidade intermitente ou cabos de par trançado deformados, pode-se experimentar sintomas como **conectividade intermitente** ou, na maioria das vezes, **rede lenta**.

Sinais: Cabeamento com continuidade intermitente ou microfissuras em rede causam **altas taxas de erros** em redes (cálculo do CRC – *Cyclic Redundancy Check*)[12]. Uma boa rede tem taxas muito próximas de zero. A probabilidade de erros em cabos de cobre sãos é de 10^{-9} e em fibras 10^{-12} (TANENBAUM, 2003).

[12] Informação anexada a um quadro na camada de enlace, com o objetivo de verificar erros durante a transmissão.

Sinais Confirmatórios: Verificação de *leds* na placa de rede e no equipamento de conectividade envolvido; teste com equipamento próprio, chamado *scanner* de rede. Este equipamento permite a verificação não só da continuidade, mas de parâmetros como NEXT (*near-end crosstalk*), atenuação e ruído. Outro equipamento útil é o TDR (*Time Domain Reflectometer*) e sua versão para fibras óticas, o OTDR, que localizam fraturas, torções, curtos, rupturas e outras irregularidades, com base no princípio em que elas provocam reflexões em sinais emitidos em um determinado cabo, com amplitude variável de acordo com o problema. O TDR mede a distância de cabo até o ponto onde a irregularidade ocorre (CHAPPELL, FARKAS, 2003); e usar o *ping* para avaliar logicamente a continuidade do cabo até o equipamento de conectividade mais próximo. Analisar o tempo de resposta e a taxa de erros dos pacotes ICMP. Para avaliação de intermitências, é conveniente deixar "pingando" o destino continuamente13 e movimentar o cabo.

Possíveis Soluções: De acordo com o resultado dos testes, cabos metálicos devem ser substituídos. Fibras podem ser reparadas, mas devem ser testadas para verificar se ainda se encontram dentro do padrão esperado para a categoria de cabeamento desejada (ANÔNIMO, 2001).

A-1.3 Conectores Irregulares

Descrição: O processo de conectorização é muito simples, mas a falta de cuidado pode resultar em pouco contato ou intermitência de contato entre os condutores. A proteção entre os condutores, as tranças, deverão inevitavelmente ser desfeitas para possibilitar a conectorização. Este alinhamento dos fios, no entanto, não deve passar de 13 mm. O tamanho exato do conector, para evitar o *crosstalk*. Outro aspecto é que para as redes mais usadas na ocasião da elaboração deste trabalho, as redes 10/100 Mbps, é necessária a troca da posição do fio 4 com o fio 6. Isto se deve ao fato deste tipo de rede usar os fios 1, 2, 3 e 6 para condução dos sinais, mas os fios são trançados aos pares, ou seja, o 1 com o 2, o 3 com o 4, o 5 com o 6. Para evitar a interferência entre os condutores usados, o fio 4 é trocado de posição com o 6. se este procedimento não for realizado, o cabo funcionará em qualquer configuração de crimpagem, desde que ambas as extremidades estejam iguais, porém poderá estar ocorrendo o

[13] Depende da versão do ping e do SO. Para Windows XP SP2, ping -t

fenômeno chamado de *split pair* (par separado), quando a interferência poderá resultar em erros e intermitência.

Sintomas: O conector mal-crimpado provoca sintomas como a intermitência de conectividade ou falta de conectividade.

- Pares separados provocam sensação de rede lenta, devido aos erros, até o limite da perda de Conectividade.

Sinais: Alguns sinais de efeitos causados por conectores mal-crimpados são os seguintes:

- Número elevado de erros são sinais de conectores mal-crimpados.
- Taxa de colisões elevada - acima de 10% (LOPES *et al.,* 2003) - podem ser ocasionadas por conectores mal-crimpados.
- NEXT e atenuações (ANÔNIMO, 2001, CHAPPELL, FARKAS, 2003) acima do esperado são indícios de *split pairs*.

Testes Confirmatórios: devem ser realizados para constatação do diagnóstico de conectores mal-crimpados. Os mais indicados são os seguintes:

- Verificação dos *leds* e inspeção visual do conector sob suspeita;
- Teste com testadores próprios, de preferência um *Scanner*.

Possível Solução: para solucionar os efeitos de conectores mal-crimpados, indica-se a troca do conector, de acordo com o padrão (ANÔNIMO, 2001).

A-1.3.1 Incompatibilidade de modo ou taxa de operação entre interfaces

Descrição: Para que as redes pudessem aumentar a taxa de operação de 10 Mbps para 100 Mbps foi necessária a implementação do modo de transmissão *full-duplex*, eliminando as colisões do método CSMA/CD e permitindo assim maior *throughput*. Para garantir compatibilidade com as redes legadas, os equipamentos tipicamente suportam ambas configurações de funcionamento, usando um par de fios em 10 Mbps ou dois pares a 10 ou 100 Mbps. Isto pode ser negociado automaticamente pelo software dos equipamentos

ou então configurada manualmente por um operador. Na prática, nem sempre a negociação automática funciona adequadamente, efeito mais notadamente percebido em equipamentos de fabricantes diferentes. Neste caso, equipamentos podem optar por operar em modos diferentes (um lado em half-duplex – 10 Mbps e outro a full-duplex – 10 Mbps ou 100 Mbps), provocando mal-funcionamento.

Sintomas: O sintoma mais comum é o da falta de conectividade com equipamentos ou serviços, quando as taxas de operação estão descasadas (uma a 10 Mbps e outra a 100Mbps). Em caso de incompatibilidade de modo de transmissão (uma a *half-duplex* e outra a *full-duplex*), o sintoma será de rede lenta.

Sinais: Os principais sinais caracterizadores de discrepâncias de configuração do modo ou taxa de operação de interfaces de rede são os seguintes:

- Número de erros alto. Qualquer coisa insistentemente acima de zero é indício de problemas na rede.
- Na operação em full-duplex, não ocorrem colisões. Caso e rede esteja apresentando colisões – taxas superiores a 10% são indicativas de problema – possivelmente há incompatibilidade entre o modo de operação de interfaces.
- Há um tipo especial de colisão chamada de "tardia", por ter ocorrido após um dos lados já ter transmitido com sucesso mais do que 512 bits de um quadro e o outro lado, por não executar o algoritmo de CSMA, inicia a transmissão provocando uma colisão.

Testes Confirmatórios: A simples verificação de modo de operação já pode permitir a verificação de incompatibilidade. Para lados configurados para automático, um software de gerenciamento pode monitorar se as configurações sobem para 100 Mbps caindo em seguida para 10 Mbps, continuamente. Isso pode ser feito através da variável *dot3StatDuplexStatus*, integrante da MIB *Ether-like* (FLICK, JOHNSON, 2006) (modo de operação – *full* ou *half*) e a variável *ifSpeed* da MIB-2 (MCCLOGHRIE, KASTENHOLZ, 2006) - taxa de operação.

Possível Solução: Deve-se optar pela configuração manual das interfaces, dando preferência pela melhor configuração possível (100 Mbps, *full-*

duplex). Isso nem sempre é possível, especialmente se há equipamentos legados, como *hubs* que operam apenas a 10 Mbps CSMA/CD.

A-1.3.2 Equipamento de Conectividade Defeituoso

Descrição: Equipamentos de conectividade, como qualquer item de hardware, são sujeitos a falhas. Descargas elétricas provocadas por variações no fornecimento de energia, raios ou curtos podem fazê-los operar inadequadamente ou até perder completamente sua operacionalidade. Tais equipamentos possuem também seu software básico, que como qualquer peça de software pode conter *bugs* que o façam operar inadequadamente. São comuns na experiência do autor situações onde a rede ou um segmento se encontra com funcionalidade reduzida ou até perda total de funcionalidade, e um simples desliga-liga (*reboot*) do equipamento de conectividade envolvido já resolve o problema. É claro que tais equipamentos possuem um MTBF que deve ser observado, para indicar se o mesmo já cumpriu sua missão, devendo então ser substituído.

Sintoma: Conforme explanado na descrição, defeitos em equipamentos de conectividade provocam sintomas de rede lenta até a perda total de conectividade.

Sinais: Os principais sinais característicos da falha causada por equipamentos de conectividade defeituosos são:

- Equipamento inoperante
- Interfaces em estado administrativo não-operacional
- Taxas de utilização de CPU e de memória acima de 75% (CHAPPELL, FARKAS, 2003) são indicativas de problemas
- Tráfego *broadcast* e *multicast* elevado. Causado por mal-funcionamento, geram as tempestades que congestionam interfaces.

Testes Confirmatórios: Devem ser utilizados para o diagnóstico preciso da localização da falha no equipamento defeituoso:

- Inspeção visual no equipamento (verificação de *leds*). Os manuais indicam situações anormais de operação.
- Verificação de status do equipamento, com ferramentas de gerência. O status das interfaces é uma das informações mais comuns nos equipamentos de

conectividade. Estas mesmas ferramentas permitem a verificação das taxas de ocupação de CPU, memória e *broadcast/multicast*.

- Testar a conectividade com o equipamento sob suspeita. O *ping* é útil para esse teste. Convém analisar não só a continuidade, mas também os tempos de resposta e as taxas de erro (perda de pacotes ICMP).
- Substituição do equipamento sob suspeita. Teste rápido e objetivo, não só mantém a rede operacional como permite que o equipamento seja analisado mais detalhadamente.

Possíveis Soluções: visando solucionar a falha, devem ser tentadas as seguintes medidas:

- A pura e simples substituição do equipamento pode resolver o problema imediatamente, porém a verdadeira causa do defeito ainda poderá estar presente e causar novas falhas. Conforme dito no início deste capítulo, o objetivo de um Gerente de Redes deve sempre ser a busca da solução correta, ou seja, que resolva o problema definitivamente e que não cause outros problemas. Após a observação do tempo de vida útil do equipamento, uma verificação no sistema de estabilização de energia elétrica e de refrigeração dos equipamentos deve ser realizada.
- As configurações dos equipamentos devem ser checadas junto às informações do fabricante e as recomendações dos manuais (*troubleshooting*).
- Atualizar o IOS dos equipamentos de conectividade é uma boa prática que deve ser realizada periodicamente, e não apenas para contingenciar problemas.

A-1.3.3 Interface de Rede Defeituosa

Descrição: Considera-se uma interface de rede não apenas uma placa de estação de rede, mas também uma porta de equipamento de conectividade. Estes equipamentos, quando defeituosos, causam um efeito bastante nocivo nas redes, porque geram ruídos que deterioram a qualidade do tráfego para as outras estações, dificultando a localização do problema. É uma das situações de mais difícil resolução, principalmente em redes grandes não-segmentadas ou mal-segmentadas. Estas interfaces podem apresentar os seguintes problemas:

- Operando em CSMA, não detecta a portadora e inicia transmissões provocando colisões e colisões tardias, explanadas na seção 3.3.3.

- Geração de quadros espúrios, principalmente de *broadcast* e *multicast*, poluindo a rede.
- Geração de quadros maiores que o MTU[14] padrão (1518 bytes).

Sintoma: Basicamente, o sintoma é o de rede lenta até a perda de conectividade, por um certo número de usuários conectados no mesmo segmento lógico que o de uma interface de *backbone* defeituosa. Se o equipamento de conectividade for um *switch* e a interface em questão for a de uma estação, o *switch* filtrará essas irregularidades e apenas o usuário desta máquina reclamará. Os quadros de *broadcast*, no entanto, poderão causar tempestades de tráfego acentuado.

Sinais: a ocorrência de defeitos em Interfaces de Rede provoca os seguintes sinais típicos:

- Taxas de erros elevadas.
- Taxas de colisões elevadas (acima de 10%) e ocorrência de colisões tardias.
- Alto tráfego de *broadcast* ou *multicast*.
- Aumento injustificado da taxa de utilização do enlace, facilmente perceptível por estações de gerência SNMP, como o MRTG (OETIKER, 2006).
- Existência de quadros superiores ao MTU padrão na rede.

Testes Confirmatórios: São realizados para assegurar que o sintoma descrito realmente pode ser diagnosticado como uma falha de interface. São os seguintes:

- Verificar se está sendo utilizado o *driver* correto da placa de rede. É muito comum a utilização de *drivers* genéricos de sistemas operacionais, que nem sempre têm a funcionalidade adequada. Na ocasião, verificar também a configuração dos protocolos de rede no SO. Há programas testadores da funcionalidade das interfaces fornecidos pelos próprios fabricantes.
- Substituir a placa de rede ou a porta do equipamento por outra e analisar a permanência ou cessação dos sinais.
- Testar a interface com *ping*, observando os tempos de resposta e a perda de pacotes.

Possíveis Soluções: Para solucionar o problema de Interface defeituosa, deve-se testar as seguintes soluções:

[14] MTU – Maximum Transport Unit – Tamanho de quadro padronizado para redes 802.3 e suas variantes

- Uma solução curiosa, mas normalmente utilizada por técnicos de rede, é a desinstalação e reinstalação do software de rede (*driver* da placa) do Sistema Operacional, em especial o Windows. Caso alguma *dll* esteja corrompida, será substituída pela original e a funcionalidade voltará.
- Verificar o encaixe físico de placas em *slots* nas estações e conectores, seguindo-se pelo teste da interface por software.
- Instalar uma versão nova (atualizada) do *driver* da interface.
- Trocar a placa ou a porta do equipamento de conectividade.

A-1.3.4 Interferências no Cabeamento Metálico

Descrição: Os cabos de par trançado possuem alguma resistência às interferências ao ruído induzido, principalmente por fontes como motores elétricos de aparelhos de ar condicionado, aspiradores de pó, reatores de lâmpadas fluorescentes e cabos de corrente elétrica, razão pela qual este tipo de cuidado é normalmente descartado. Acima de uma certa potência, no entanto, tais fontes de ruído deterioram o sinal e são de difícil detecção. Outra causa importante de ruído são fontes de radiofreqüência.

Sintomas: Ruídos causam tipicamente a sensação de rede lenta, na maioria das vezes experimentada apenas pela estação atendida por um cabo passando próximo às fontes de interferência. Na maioria das vezes, esta sensação é intermitente, ocorrendo quando o compressor do ar-condicionado é acionado, ou a lâmpada é acesa.

Sinais: O ruído causa erros que podem ser detectados através de ferramentas de gerência no cômputo de estatísticas de erros de CRC. Como já citado anteriormente, devem-se esperar taxas de erros muito próximas de zero.

Testes Confirmatórios: O teste mais eficaz e prático é o acompanhamento da rota do cabo para verificar possíveis fontes de interferência. Desligando ou afastando as mesmas e observando as taxas de erro pode-se confirmar a interferência.

Possível Solução: A solução mais razoável é lançar outro cabo por um caminho distante das fontes de interferência.

A-1.3.5 Congestionamento em Barramentos

Descrição: A operação em topologia lógica de barramento pressupõe a operação em modo *half-duplex*, ou seja, há compartilhamento de cabos para

enviar e receber quadros, não sendo tolerada a simultaneidade. Para que haja ordem no acesso ao meio físico, é utilizado o método CSMA/CD, quando uma estação só transmite quando o meio físico está desocupado (*Carrier Sense Multiple Access*). Podem, no entanto, ocorrer colisões causadas pela coincidência na verificação do meio livre, daí a utilização do CD (*Collision Detection*). Neste processo, as estações em estado de colisão aguardam um tempo aleatório exponencialmente crescente, até uma nova tentativa de retransmissão. É fácil perceber que, num ambiente congestionado, o número de colisões fique elevado e a performance da rede seja reduzida.

Sintomas: A sensação para usuário é a de rede lenta

Sinais: Taxas de colisões elevadas (acima de 10%) podem ser observadas através de ferramentas de gerência.

- Taxa de ocupação de banda nos segmentos *half-duplex* superior a 50% CHAPPELL, FARKAS, 2003). Em modo de operação *full-duplex*, taxas de até 70% são suportáveis sem prejuízo para a operacionalidade da rede.

Testes Confirmatórios: Para verificar a compatibilidade com o perfil de falha "congestionamento no barramento", deve-se obter as seguintes informações:

- Identificação da origem das colisões.
- Identificação de estações que possam estar provocando saturação, inundando o barramento com *broadcasts* desnecessários.

Possíveis Soluções: de acordo com os testes confirmatórios, deve-se testar uma das seguintes soluções possíveis

- *Hubs* são equipamentos que implementam um domínio único de colisões. Estes devem ser substituídos por *switches*.
- Redes congestionadas devem ser reprojetadas e segmentadas, seja através de roteadores e subredes ou através de VLANs.

A-1.3.6 Utilização Incorreta de Cabos

Descrição: O detalhamento de uma categoria de componentes de cabeamento (cabos, conectores, tomadas, etc.) descreve uma série de características que, se seguidas, devem permitir o alcance de parâmetros mínimos de freqüência, atenuação e NEXT, o que, por sua vez, poderá garantir o alcance de uma determinada taxa de transmissão. Distâncias máximas de cabos

também devem ser respeitadas, sob pena de ocorrência de colisões não detectadas (em caso de transmissão *half-duplex*) e perda de dados por excesso de atenuação (em transmissão *full-duplex*). Outro aspecto é a conectorização dos cabos. Para interligar estações a equipamentos de conectividade, usa-se cabo *straight-thru* (direto) e para interligar equipamentos ou estações aos pares, usa-se cabo *crossover* (cruzado).

Sintomas: Os seguintes sintomas podem ser declarados pelos usuários:

- Caso as conectorizações estejam inadequadas para a interligação desejada, haverá falta de conectividade.
- Caso o cabeamento não esteja de acordo com a qualidade necessária para o alcance das taxas desejadas, a rede pode aparentar lenta, não operar na taxa compatível com a placa e a interface do equipamento ou ainda perder a conectividade.

Sinais: Cabos em desacordo com as especificações apresentam taxas de erro, especialmente erros de alinhamento.

Testes confirmatórios: Para confirmar a existência de discrepâncias no cabeamento, deve-se testar o seguinte:

- Verificação visual dos *leds*
- Verificação visual das extremidades do cabo para detectar seu tipo (direto ou cruzado)
- Verificar a categoria do *cabling*, através de sua certificação com equipamento adequado (*scanner*).

Possíveis soluções: No caso de cabeamento com problemas pode-se testar uma das seguintes opções de solução:

- Para cabos conectorizados inadequadamente, reconectorização ou troca do cabo
- Para inadequação de categoria, substituir cabos, conectores, caixas, etc. em desacordo com o desejado e certificar o mesmo. Para cabos muito longos, re-rotear o mesmo para alcançar o tamanho adequado.

A-1.3.7 Descumprimento de Regras do Padrão Ethernet

Descrição: Com a popularização das redes, é comum a disponibilidade de dispositivos de conectividade de baixo custo nos *shoppings* de informática, motivando usuários a criarem seus próprios ambientes de rede em casa ou

pequenos escritórios. Tal fenômeno criou uma legião de "especialistas" em redes, que levam esta prática para as corporações, implementando sua próprias soluções sem o conhecimento do Gerente da Rede quando precisa de mais pontos de rede ou mudar-se para uma sala mais distante do *wiring closet* local. Durante o projeto de uma rede, o especialista cuida da conformidade com uma série de regras descritas nas normas IEEE 802 e EIA/TIA (ANÔNIMO, 2001), que especificam distâncias máximas, afastamentos mínimos de fontes de ruído, número máximo de estações por domínio de colisões, cascateamento máximo, entre outras. Outro aspecto relevante, porém que não é o foco deste trabalho, é a questão da segurança, uma vez que a norma prevê a alocação dos dispositivos em *racks* convenientemente alocados em posições de acesso restrito, regra que é quebrada quando se cascateia uma porta de *switch* para um outro *switch* que fica em local de circulação pública.

Sintomas: Curiosamente, os usuários que ignoram os procedimentos de gerência de redes são os primeiros a relatar lentidão na rede. Em algumas situações, pode-se também chegar à perda de conectividade.

Sinais: Alguns sinais típicos do descumprimento das normas de cabeamento são os seguintes:

- Colisões em excesso (superior a 10% do tráfego) e taxa de utilização da rede muito elevada podem ser indicativos de excesso de estações no domínio.
- Colisões tardias podem ser observadas quando cabos estão maiores que o recomendado. Este sinal também é observado quando a regra de cascateamento máximo é descumprida.
- Atenuação excessiva, medida com *scanners*, é indicativa de cabo muito longo.

Testes Confirmatórios: Verificar, à luz da documentação topológica da rede, se há trechos criados pelos usuários. Caso haja, conferir se os mesmos introduzem violações das regras adequadas para o tipo de rede.

Possível Solução: Corrigir a discrepância da forma mais adequada de forma a cumprir as regras.

A grande maioria dos problemas de uma rede ocorre no nível físico, e se encaixam em uma ou mais situações citadas nesta seção. Na próxima seção serão descritos alguns problemas do nível de enlace, menos comuns, porém ainda responsáveis por algumas situações de falhas típicas.

A-1.4 Problemas de Enlace de Dados

O nível de enlace, responsável pelo método de acesso ao meio físico e pelo controle de erros em quadros, introduz maior complexidade no gerenciamento uma vez que lida com endereçamento e algoritmos que normalmente não são de conhecimento dos técnicos instaladores de redes. Esta seção abordará algumas das falhas típicas que podem causar indisponibilidade de recursos da rede.

A-1.4.1 Interface Administrativamente Desabilitada

Descrição: Com o uso de ferramentas de gerência SNMP é possível alterar o estado administrativo de uma interface de estação ou equipamento, desabilitando-a. Esta prática pode ser útil para o controle topológico da rede, dificultando que usuários façam expansões próprias usando portas livres. Outra vantagem é a de poder desabilitar remotamente a porta de uma estação que esteja gerando quadros espúrios para a rede, até que o problema seja solucionado. Ao se precisar usar a porta em questão, é possível que não se recorde que a mesma se encontra desabilitada, resultando numa ocorrência de falha. Fragilidades na segurança dos equipamentos de conectividade, como o uso de senhas *default*, pode permitir também que invasores desabilitem interfaces.

Sintoma: Falta de conectividade

Sinais: para enriquecer o caso, os seguintes sinais devem ser buscados para diagnóstico de Interface administrativamente desabilitada:

- Apesar da interface "responder" aos testes de "vitalidade", o tráfego tanto de transmissão quanto de recepção é zero.
- Com o uso de uma ferramenta de gerência, verifica-se que o estado administrativo da interface é "desabilitado".

Testes confirmatórios: A confirmação do estado administrativo da interface como desabilitada é suficiente para o diagnóstico diferencial.

Possível Solução: Alterar o estado administrativo da interface com a ferramenta SNMP adequada.

A-1.4.2 Tempestades de Broadcasts[15]

Descrição: Este é um problema muito típico em redes, uma vez que os principais protocolos de rede usam quadros de *broadcast* para várias

[15] Broadcast é um quadro ou mensagem de processamento obrigatório por todas as estações pertencentes a um mesmo domínio.

funcionalidades, e a maioria das redes usa várias arquiteturas simultaneamente. Redes que operam com IPX/SPX, TCP/IP e NetBIOS/NetBEUI, por exemplo, estão no limiar da utilização de *broadcast*, muitas vezes para uma mesma funcionalidade. Este é o caso da descoberta de nós da rede, feita em *broadcast* pelas três arquiteturas. Caso o domínio de *broadcasts* não esteja adequadamente dimensionado, poderão ocorrer situações em que as estações fiquem permanentemente ocupadas processando estes quadros. Os domínios de *broadcasts* são definidos logicamente, e podem ser segmentados com o uso de roteadores ou pelo estabelecimento de VLANs com comutadores. Uma rede não-segmentada representa um único domínio de *broadcasts*, e por representar a situação mais comum, este tipo de problema é freqüentemente observado.

Sintomas: Conforme explanado na descrição, uma rede, experimentando uma tempestade de *broadcasts* fica lenta até a perda total de conectividade. Técnicos de rede costumam desligar todos os equipamentos do *backplane* da rede por alguns minutos, ao observarem os *leds* de todos os equipamentos piscarem muito rapidamente e em aparente sincronismo.

Sinais: São sinais de ocorrência de uma tempestade de *broadcasts*:

- Com o uso de um analisador de protocolos, é trivial obter a taxa de *broadcasts* na rede. Uma avaliação da razoabilidade da taxa aferida pode ser feita observando que o tráfego normal introduz no domínio um *broadcast* por estação a cada 10 segundos (CHAPPELL, FARKAS, 2003).
- A necessidade de processar todos os quadros de *broadcasts* recebidos faz aumentar a taxa de utilização da CPU. Uma ferramenta de gerência como o MRTG pode permitir a monitoração contínua deste parâmetro.
- A recepção destes quadros por *switches* provoca a sua replicação para todas as portas, fazendo aumentar a taxa de utilização dos enlaces. O MRTG também permite a monitoração contínua deste parâmetro.

Teste Confirmatório: A observação do aumento desproporcional da taxa de *broadcasts* é sinal diferencial. Cabe ressaltar que esse problema pode ser artificialmente provocado por um ataque à disponibilidade do tipo *deny-of-service*.

Possível Solução: A melhor solução para um domínio de *broadcasts* congestionado é a sua segmentação. É recomendável também uma detalhada análise da necessidade real necessidade de múltiplas arquiteturas de rede

diferentes. A simples eliminação de uma delas já melhorará sensivelmente o tráfego. O autor recomenda a eleição da arquitetura NetBIOS/NetBEUI para eliminação, haja vista que os Sistemas Operacionais de servidores e estações Windows ao tempo de redação deste trabalho já suportam a operação baseada exclusivamente em TCP/IP. O IPX/SPX, uma arquitetura muito utilizada para implementação de servidores de arquivos no final da década de 90, hoje é representativa principalmente de sistemas legados. Na impossibilidade de unificar uma solução baseada exclusivamente em TCP/IP, convém realizar a filtragem do *broadcast* IPX nos equipamentos de conectividade, limitando o seu alcance apenas onde for absolutamente necessário.

A-1.4.3 Inadequação de Tratamento de Entradas em Switches

Descrição: No nível de enlace, *switches* gerenciam o endereçamento de quadros nó-a-nó, reduzindo o tráfego da rede por tornar *unicast* o que nas redes de barramento com *hubs* é intrinsecamente *broadcast*. Para tornar essa função factível, estes equipamentos manipulam uma tabela dinâmica de endereços físicos (MAC) em memória volátil. Estes endereços são alcançáveis através das respectivas interfaces (portas) do *switch*. Essa tabela de correspondência endereço *versus* porta é criada principalmente através da recepção de quadros, quando o *switch* "aprende" e armazena o endereço de origem. Quando, no entanto, o comutador não possui uma entrada para a máquina de destino, ele envia o quadro em *broadcast*. Em grandes redes, esse procedimento poderá inundar a rede caso ocorra freqüentemente. Para evitar isso, as entradas na tabela possuem um tempo de envelhecimento (*aging time*), antes do qual a entrada não fica obsoleta. Após isso, a mesma é descartada por não ter sido mais usada, indicando uma possível mudança de topologia. Por outro lado, manter entradas por tempos muito dilatados na tabela pode causar despachos incorretos de quadros.

Sintomas: Com tempos de manutenção das entradas na tabela muito curtos, a rede pode aparentar lenta, e com tempos muito longos a conectividade pode ficar intermitente até que a tabela seja atualizada.

Sinais: O tempo curto provocará tempestades de *broadcasts* que provocarão efeitos visualmente perceptíveis nos *switches* através dos *leds* indicativos do tráfego.

Testes Confirmatórios: Essa inadequação pode ser aferida com o uso de uma estação de gerência SNMP, através da verificação da variável *dot1AgingTime* (DECKER *et al.,* 2006).

Possível Solução: Ajuste do tempo de envelhecimento nas tabelas para o valor default, 300 segundos. Posteriormente, ajustar este tempo e monitorar para busca do valor ideal.

A-1.4.4 Inadequação de Tratamento de Entradas em Cache ARP

Descrição: Situação semelhante à descrita na seção anterior, ocorre com estações em redes *ethernet* e TCP/IP, onde o alcance de estações depende da descoberta do endereço de enlace (MAC) correspondente ao endereço lógico (IP) da estação de destino. Para descobrir uma resolução que ainda não possua, é usado o protocolo ARP, enviado em *broadcast*, onde se solicita a um determinado endereço IP que declare o seu endereço MAC. Estas associações são armazenadas em uma *cache*, que se não estiver configurada com tempo de envelhecimento adequado provocará efeitos semelhantes aos da seção anterior. O uso do DHCP e trocas freqüentes de placas de rede podem agravar o problema, se o tempo de vida das entradas estiver muito prolongado, causando despachos incorretos. Tempos muito curtos, por sua vez, provocam tempestades de *broadcasts*.

Sintomas: O problema de inadequação dos tempos de Vida das entradas em tabela é tipicamente descrito através dos seguintes sintomas:

- Com o tempo de validade muito curto, a rede aparentará lenta devido aos *broadcasts* excessivos.
- Com o tempo muito extenso, poderão ocorrer situações de aparente falta de conectividade quando um IP ou MAC for substituído numa estação, até que a entrada obsoleta expire e seja substituída por outra correta.

Sinais: O número típico de quadros ARP em uma rede por segundo deve ser em torno do número de máquinas existentes num mesmo domínio de colisão (CHAPPELL, FARKAS, 2003). Valores superiores a isso devem ser investigados.

Testes Confirmatórios: Com o uso de um analisador de protocolos, verificar a taxa de *broadcasts* e a origem de *broadcasts* em excesso. De acordo com o Sistema operacional em uso, conferir o tempo de validade da *cache* ARP.

Possível Solução: Ajuste do tempo de vida da cache ARP para o valor default, 600 segundos (LOPES *et al.,* 2003). Posteriormente, ajustar este tempo e monitorar para busca do valor ideal.

Esta seção abordou alguns dos problemas mais típicos do nível de enlace. A seguir, serão selecionados alguns problemas mais genéricos do nível de rede, passíveis de estarem presentes em qualquer instalação.

A-1.5 Problemas da Camada de Rede

A camada de Rede, responsável pelo controle do roteamento na rede, o endereçamento lógico das estações e a criação dos pacotes, entre outras funções, possui os seguintes problemas típicos:

A-1.5.1 Configuração de Default Gateway Incorreto

Descrição: Roteadores são equipamentos necessários para a interconexão de redes locais. São logicamente posicionados entre as redes, exercendo a função de *gateway*[16] para as mesmas. Caso um pacote não seja endereçado a alguma máquina no escopo interno da rede, ele é direcionado para uma outra interface do roteador, através da qual endereços externos possam ser alcançados. Este processo se inicia no *host* de origem, através da verificação do endereço e da máscara de destino. Caso a rede de destino, calculada através da operação lógica binária AND entre o endereço e a máscara resulte em um endereço de rede diferente do pertencente à máquina de origem, o pacote é despachado para o *default gateway*, previamente configurado em toda estação de rede, que se encarregará de descobrir a melhor rota para alcançá-lo. Este endereço deve permanecer corretamente configurado, para que as comunicações com máquinas externas à rede possa acontecer corretamente.

Sintomas: A comunicação entre a máquina e os servidores internos e outras máquinas da própria rede ocorre normalmente, mas não há conectividade com o mundo externo. Outras máquinas pertencentes à mesma rede funcionam normalmente.

Sinais: Quando um *host* numa rede não possui uma rota correta para alcançar um destino, o roteador enviará para este *host* mensagens ICMP de

[16] Elemento de interface entre ambientes heterogêneos no que diz respeito ao endereçamento e, possivelmente, tecnologia de rede. Elemento de passagem obrigatória de todos os pacotes destinados a outras redes.

redirecionamento, para informá-la que existe um caminho disponível para alcançar o destino desejado.

Testes Confirmatórios: Os seguintes testes podem ser realizados para confirmar o diagnóstico da falha.

- Verificação da configuração do *default gateway*.
- Verificação, com o uso de um analisador de protocolos, da existência de mensagens de ICMP *redirection* para a máquina reclamante.

Possível Solução: Configuração correta do *default gateway*.

A-1.5.2 Endereço IP incorreto

Descrição: Por mais que possa parecer incomum, a ocorrência desse erro é típica, principalmente se a rede for segmentada em subredes e o erro for na digitação, dentro do escopo de uma subrede, de um endereço pertencente a outro domínio de *broadcasts*.

Sintomas: A configuração de um endereço IP duplicado é de diagnóstico trivial, uma vez que os principais Sistemas Operacionais já indicam tal ocorrência durante o *bootstrap*. Caso o endereço esteja incorretamente configurado para o seu escopo de subrede (ou rede), não haverá alerta de erro, mas não haverá conectividade.

Sinais: A utilização de um Analisador de Protocolos permite a verificação de pacotes com endereços de origem não-pertencentes a subrede em teste, especialmente com destino *broadcast*.

Testes Confirmatórios: A verificação do endereço configurado na máquina identificada com o Analisador de Protocolos confirma ou não a ocorrência da discrepância. Uma das ferramentas desenvolvidas para operação como *plug-in* neste trabalho permite a realização desta tarefa pelo próprio usuário.

Possível Solução: Configuração correta do endereço e reinicialização do Sistema Operacional. Convém testar a conectividade da máquina com o *ping* para eliminar outras possíveis falhas.

A-1.5.3 Host com Máscara de Rede Incorreta

Descrição: A principal finalidade da máscara é definir a qual domínio uma interface pertence. Uma operação lógica *AND* entre os bits de um endereço e da sua máscara resulta no endereço da rede (ou subrede) a qual o mesmo

pertence. Caso a máscara seja digitada incorretamente, resultará num contexto de domínio maior ou menor do que o esperado, o que provocará uma falha curiosa. Caso o domínio configurado seja maior que o real, a interface interpretará como internos alguns despachos que na verdade se destinam a outros domínios, e, caso a máscara defina um domínio menor, ela despachará para o *default gateway* pacotes que, na verdade, são internos.

Sintomas: Em ambos casos a sensação percebida será a de intermitência na conectividade, especialmente no acesso a algumas máquinas ou serviços.

Sinais: Caso o domínio tenha sido incorretamente reduzido através da máscara, a máquina reclamante receberá mensagens de ICMP REDIRECT do *Default Gateway*, indicando que a entrega poderia ser direta.

- Se o domínio for equivocadamente ampliado, trafegarão pela rede requisições ARP, buscando endereços MAC para possibilitar entregas diretas, sem resposta.

Testes Confirmatórios: A verificação da configuração do endereço/máscara confirma a falha. Remotamente, pode-se usar um analisador de protocolos para capturar os sinais indicados na seção anterior.

Possível Solução: Reconfiguração do endereço/máscara e reinicialização do Sistema Operacional. Convém testar a conectividade da máquina com o *ping* para eliminar outras possíveis falhas.

A-1.5.4 Configuração do Cliente DNS incorreta

Descrição: O DNS (*Domain Name System*) é de fundamental importância para os serviços de rede (TANENBAUM, 2003). Para possibilitar o elevado nível de abstração do usuário quanto à infra-estrutura da rede, questões de endereçamento e roteamento são ocultadas do usuário através do uso de nomes, fáceis de memorizar e de associar aos serviços desejados. Para que este esquema funcione, é necessária uma estrutura hierárquica de servidores de nomes, que os transformam em endereços para os clientes requisitantes. Caso o cliente DNS não esteja configurado ou esteja com endereço errado, tais resoluções não ocorrerão, impossibilitando o acesso aos serviços de rede.

Sintomas: O usuário reclamará de falta de conectividade.

Sinais: Os serviços funcionam normalmente se forem acionados através do endereço IP, ao invés do nome.

Testes Confirmatórios: A aplicação desenvolvida para este trabalho permite a verificação local da configuração do DNS.

Possível Solução: Configuração correta do Cliente DNS.

Esta seção ilustrou algumas das situações mais típicas encontradas nas redes, cuja ocorrência de falha se enquadre na camada de rede do modelo OSI. A próxima seção se limitará a descrever uma ocorrência bastante usual atualmente, a contaminação viral.

A-1.6 Problemas da Camada de Aplicação

Os problemas da Camada de Aplicação estão ligados aos programas utilizados na camada mais externa da pilha de protocolos, com a qual o usuário tem contato direto. Tal situação permite uma excessiva liberdade de alterações no ambiente de produção, que introduz algum nível de vulnerabilidade. A larga utilização atual, por exemplo, de computadores para comércio eletrônico e *banking*, fez surgir uma categoria específica de marginais, interessados em capturar senhas de acesso que lhes traga alguma vantagem. Em função disso, apesar da possibilidade de ocorrência de outra sorte de falhas, esta seção se concentrará apenas no malware[17].

A-1.6.1 Contaminação Viral do tipo worm

Descrição: Uma máquina contaminada com vírus pode manifestar esta ocorrência de diversas formas. Para os interesses deste trabalho, serão focadas apenas as técnicas cuja estratégia de propagação provoque queda de desempenho na rede. Neste caso, uma vez contaminada, a máquina buscará a propagação do seu código para outras máquinas da rede, usando alguma falha existente no Sistema Operacional. Sem que o usuário perceba, a máquina testa conectividade com as demais do mesmo range de endereços IP, tenta abrir conexões de transporte com estas máquinas e transmitir o seu código. Essa busca incessante de conexão com outras máquinas provoca grande saturação da rede, sendo percebida por todos os usuários da rede, estejam no mesmo barramento lógico ou não. A seção 6.1 ilustrou a situação descrita.

Sintomas: O sintoma predominante, não apenas para a máquina contaminada como para toda a rede, é o de Rede Lenta.

[17] *Malware* é um tipo de código que busca realizar alguma tarefa que poderá provocar prejuízo para outrem. É tipicamente disseminado eletronicamente, anexo a e-mails ou propagado via rede, usando vulnerabilidades dos Sistemas Operacionais.

Sinais: A rede com uma ou mais máquinas contaminadas sofre com os *broadcasts* do tipo ARP, que causa saturação. Os *leds* piscam incessantemente, e, mesmo desligando-se e religando os equipamentos de conectividade a lentidão volta a ocorrer.

Testes Confirmatórios: O uso de um *sniffer* permite a observação dos seguidos ARP enviados em broadcast seqüencial. A máquina de origem destas mensagens, se retirada da rede ou desligada, cessa imediatamente os sintomas e sinais. Uma análise desta máquina, em operação normal, pode ter várias conexões de transporte abertas, para proliferação do código malicioso.

Possível Solução: Pesquisa nos repositórios especializados da melhor forma de eliminar o vírus. Após a descontaminação, atualizar o Sistema Operacional da máquina quanto às falhas de segurança. Instalar um antivírus com atualizações automáticas e inspeção de arquivos antes do *download*, armazenamento e da execução.

Este apêndice detalhou os problemas típicos do ambiente das Redes Locais de computadores. Naturalmente, não esgota a diversidade de situações de falha, mas atende ao propósito de inserir na aplicação CBR os casos mais freqüentes, para aproveitamento da experiência especialista disponível na literatura e reduzir o tempo de perda de disponibilidade das estações de trabalho. Naturalmente, novas falhas, menos comuns e eventualmente particulares para o ambiente de uso da aplicação CBR, serão inseridas aumentando a eficiência da equipe de suporte no diagnóstico e solução dos problemas na Rede Local.

Apêndice 2 Algumas Telas da Documentação da Aplicação CBR

Neste apêndice, algumas telas da documentação gerada para o desenvolvimento e a utilização da aplicação CBR desenvolvida são descritas. Esta documentação foi elaborada em conformidade com a metodologia INRECA II, amplamente discutida na literatura e que atendeu plenamente aos objetivos do trabalho.

O padrão INRECA prevê a criação e disponibilização das fichas de documentação em ambiente hipertexto, facilitando assim a navegação entre os diversos passos da construção, implementação e testes do sistema. Esta abordagem permite a rápida consulta por todos os envolvidos no projeto. Apenas para ilustrar os resultados desta fase do trabalho, algumas telas são descritas a seguir;

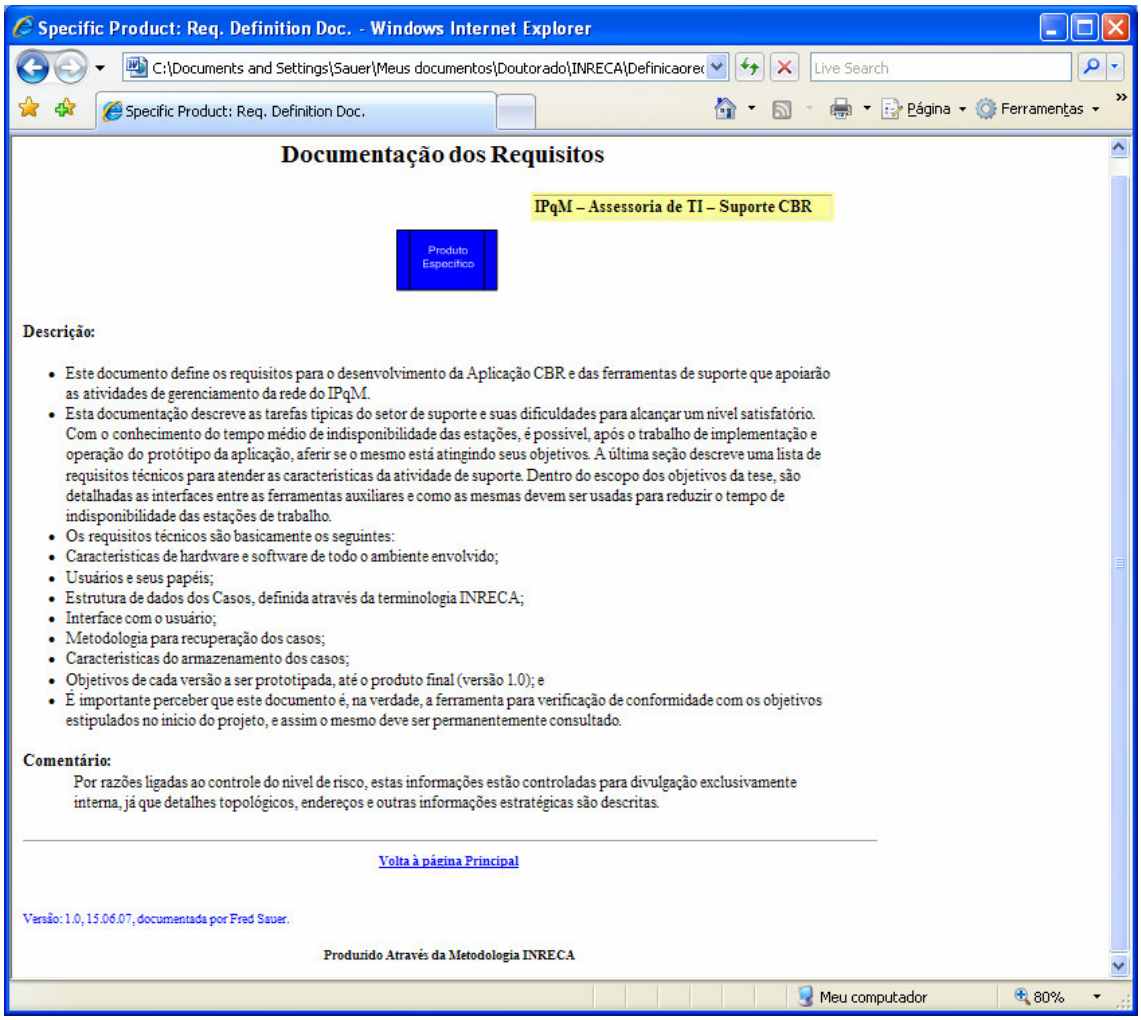


Figura 67- Produto Específico "Documentação de Requisitos"

Na figura 67, é ilustrada a descrição das características da documentação dos requisitos iniciais para o desenvolvimento da aplicação CBR. Por tratar-se de documento com teor sensível, por conter informações sobre topologia e endereçamento da rede do IPqM,

apenas linhas gerais são descritas, mantendo-se os detalhes sigilosos em documentação física.

Figura 68 - Produto Específico "Aquisição de Conhecimento"

A figura 68 ilustra a definição do produto "Aquisição do Conhecimento", quando é documentada a estratégia adotada para obter o material inicial da formação da Base de Casos da aplicação CBR. Cabe ressaltar a importância da informação acerca de números de casos usados, que além de servir para o aperfeiçoamento do sistema em discussão, dá a outros interessados na adoção do CBR uma orientação de como estipular metas iniciais que, comprovadamente, trazem resultados para um ambiente similar.

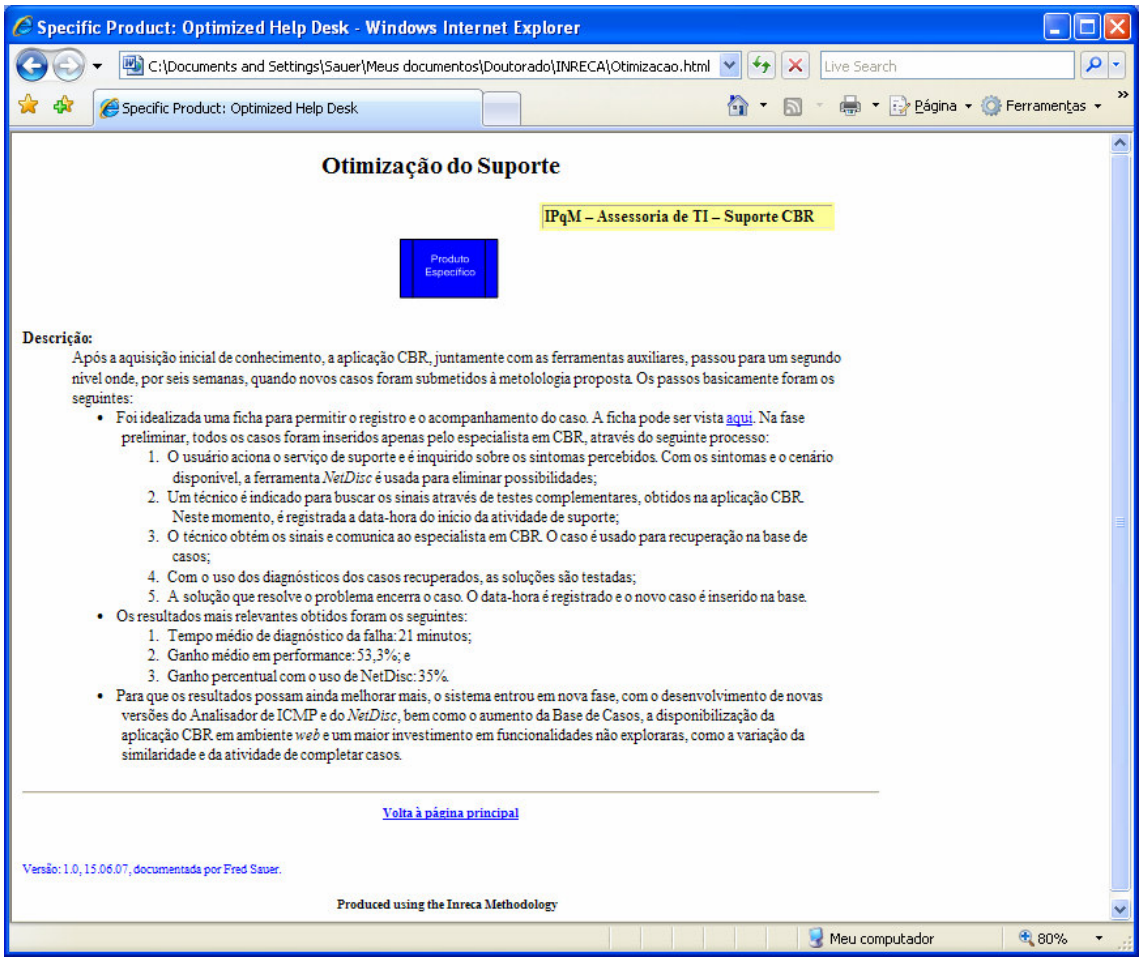


Figura 69 - Produto Específico "Otimização do Suporte"

A figura 69 apresenta a captura da tela, em ambiente *web*, da ficha de documentação referente ao produto específico “Otimização do Suporte”. Nesta ficha, é descrita a metodologia (processo) de utilização da aplicação CBR e das ferramentas auxiliares, os resultados obtidos mais relevantes e, por fim, os próximos passos a adotar. Esta estratégia de documentação permite um acompanhamento dinâmico da evolução do sistema, além de tornar ampla a divulgação de detalhes fundamentais para padronização de procedimentos, verificações de conformidade e, em última análise, o sucesso da abordagem para melhoria da performance do pessoal de suporte e, consequentemente, redução do tempo de indisponibilidade das estações de trabalho.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo