

Detecção e análise de falhas usando redes de dependências em sistemas distribuídos de gerenciamento

João Carlos F. Barbosa, José Marcos S. Nogueira
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
{joca,jmarcos}@dcc.ufmg.br

Resumo

Para acompanhar um grande crescimento do número de equipamentos gerenciados, aumento no número de operações, dados tratados ou mesmo significativa expansão da área geográfica, um sistema distribuído de gerenciamento precisa apresentar grande flexibilidade e escalabilidade. Muitas vezes falhas ocorrem em decorrência destes fatores e, devido aos muitos módulos e recursos utilizados e das complexas relações de dependência entre eles, explícitas ou não, torna-se difícil isolar a causa principal das falhas.

Este trabalho apresenta uma aplicação de técnicas de modelagem de sistemas com redes de dependência, com o objetivo de possibilitar e facilitar a detecção de falhas neles ocorridas. A partir da identificação de modelos e caracterização de dependências, foi modelado um sistema distribuído de gerenciamento com o objetivo de se construir uma representação para facilitar o diagnóstico dos problemas. Essa técnica baseia-se na extração automática de dependências definidas nos modelos de representação e em dados de configuração dos sistemas. O resultado da técnica é um grafo estrutural de dependências e um roteiro para auxiliar a identificação de problemas.

Foi utilizado como motivação e ambiente de aplicação da técnica o Sistema Integrado de Supervisão, um sistema distribuído de gerenciamento de redes.

Palavras-chave

Gerenciamento de redes de computadores e de telecomunicações, redes de dependência, diagnóstico de falhas, Sistema Integrado de Supervisão

Abstract

A distributed management system must present flexibility and scalability in order to follow growing on number of managed devices, increasing on number of operations, data handled, or even expansion of its geographical area. Much of the time problems occur caused by such growing factors and, because the large number of modules and resources as well as the complex relationships among them, it is difficult to isolate the main cause of the problem.

This paper presents an application of system's modeling techniques using dependencies networks targeting facilitate problems detection in such systems. Starting from models identification and dependency characterization, a distributed management system has been modeled to construct a representation to facilitate diagnosis of the root-cause. This technique is based on automatic extraction of dependencies models and on configuration data. The result is a dependency structural graph and a script to help in problem identification.

It was used as motivation and test bed of the techniques the SIS platform, a distributed management system used in telecommunication plants.

Keywords

System to Integration of the Supervision, network management, dependency networks, fault diagnosis

1. Introdução

O rápido crescimento das plantas de telecomunicações, tanto em extensão, em número de equipamento quanto em diversidade de tecnologia e fabricantes, traz um significativo aumento da complexidade de gerenciamento dos equipamentos que compõem estas plantas. Para tratar estes problemas, sistemas de gerenciamento distribuído de tais redes de telecomunicações são compostos por vários módulos distintos que se comunicam e interagem entre si, além de módulos para a comunicação com os diversos equipamentos gerenciados.

Tipicamente tem-se sistemas distribuídos com módulos de alto grau de interdependência que se comunicam basicamente utilizando protocolos de sessão, transporte e rede (RPC e TCP/IP). Nesse tipo de sistema a distribuição dos módulos comunicantes é muito variável, desde a comunicação entre processos em uma mesma máquina, passando pela comunicação em rede local, até a comunicação remota, em redes de longa distância.

Entender a dinâmica da comunicação e o inter-relacionamento entre tais módulos é uma tarefa complexa. Prever falhas, ou mesmo detectá-los a partir de sintomas fornecidos é uma tarefa ainda mais difícil. Diversos problemas podem ocorrer com os módulos neste tipo de arquitetura, indo da comunicação entre eles até mesmo à infra-estrutura necessária para seu funcionamento.

Através da construção de redes de dependências entre os módulos desse tipo de sistema, podemos mais rapidamente isolar as causas de problemas, e tomar as ações

apropriadas para resolvê-los. Em uma plataforma de gerência de falhas, modelos de dependências podem ser utilizados ainda como ferramenta para a correlação dos alarmes coletados ou percebidos.

Alguns trabalhos podem ser encontrados que utilizam a construção de redes de dependência no gerenciamento de falhas, geralmente com foco na correlação de alarmes. Outros tentam determinar automaticamente as dependências, e gerar mapas com graus de dependências entre componentes, por exemplo em sites de comércio eletrônico.

A contribuição proposta neste trabalho consiste numa técnica para determinar redes de dependências em sistemas distribuídos, gerando-as automaticamente a partir de um modelo básico, e da configuração do sistema em um banco de dados. Ainda, utilizar essas redes para criar formas eficientes de detecção e correção de problemas nos sistemas, bem como criar roteiros de ação para ajudar operadores e administradores do sistema a isolar falhas e acionar as equipes apropriadas para solucioná-las.

Usou-se como referência o Sistema Integrado de Supervisão – SIS [NOG96], como motivador e como ambiente para demonstrar os conceitos aqui tratados. Dado o recente crescimento do SIS, em vista da expansão da área de utilização (planta da Telemar), alguns problemas começaram a surgir, e suas causas demonstraram-se de difícil resolução. Além disso, é previsto um aumento de uma ordem de grandeza na carga de trabalho exercida sobre o sistema em algumas regiões, o que traz certa preocupação sobre o comportamento do sistema.

Desta forma, a proposta deste trabalho mostra-se de grande importância para uma compreensão deste comportamento, bem como para a criação de mecanismos automatizados de medição e avaliação de desempenho, além de previsão e detecção de problemas.

O presente texto é organizado como se segue. Na seção 2 são apresentados modelos de dependência e trabalhos relacionados, usados em diagnósticos de falhas e correlação de alarmes. É também apresentada uma caracterização de dependências e formas de representação. Ao final é apresentado um método para a geração automática de uma rede de dependências, a partir de um modelo funcional, e a geração de um roteiro de diagnóstico de falhas para o administrador do sistema, a partir desta rede. Na seção 3 é apresentada a estrutura do SIS, e seu processo de gerenciamento de falhas. Na seção 4 é realizada a modelagem de redes de dependências do SIS, de acordo com modelos apresentados nas seções anteriores e apresentada a aplicação dos conceitos desenvolvidos ao mesmo.

2. Modelos de dependência no diagnóstico de falhas

Existem diversas referências na literatura sobre o uso de modelos de dependências no diagnóstico de falhas e procura da causa principal (*root cause analysis*). Entretanto duas abordagens se destacam. A primeira diz respeito ao problema de correlação de alarmes, como descrito em [YEM96], [CHO99] e [GRU98]. Nesse tipo de sistema, alarmes ou eventos recebidos são inicialmente mapeados nos nós de um grafo de dependências correspondendo às origens dos eventos. As dependências destes nós são examinadas para identificar o conjunto de nós com eventos ou alarmes que potencialmente serão as causas do evento ou alarme percebido.

A outra abordagem do uso de modelos de dependências na análise de falhas é utilizar grafos para realizar um exame sistemático do sistema em busca da causa principal do problema. Métodos estatísticos ou probabilísticos podem ser utilizados para realizar a busca primeiro nos módulos com maiores probabilidades de serem os causadores do problema. Essa abordagem está descrita em [KAT97].

Uma fonte de preocupação está relacionada às dificuldades de elaboração de tais redes de dependências. Em geral é uma tarefa árdua, que consome tempo e requer um grande conhecimento dos sistemas. Por estes fatores, a construção das redes também está sujeita a falhas e partes importantes das dependências podem ser omitidas. Em [ENS99] e [KAR00] este problema é abordado, onde os autores sugerem metodologias para a extração de dependências.

Em [KEL00] é feita uma classificação das dependências e em [KAR00] os mesmos autores propõem uma técnica automática para tratar o problema da geração das dependências dinâmicas, que são difíceis de se prever manualmente. Em [BRO01] é apresentado o conceito de descoberta ativa de dependências através da perturbação e monitoração automática do sistema. Já em [THO01] é apresentado uma metodologia para a tomada de ações orientada a eventos num sistema de gerenciamento de eventos.

Dentro do próprio Sistema Integrado de Supervisão, a questão da correlação de alarmes foi amplamente discutida em [MEI00] e [MEI97], onde neste último encontra-se um apanhado completo sobre o assunto até então. Em [SAF99] os autores apresentam um método para a correção automática de falhas pelo SIS, através do emprego de raciocínio baseado em casos.

2.1. Caracterização de dependências

Pode-se definir uma dependência como a expressão de como o funcionamento de um componente afeta o funcionamento de outro. Em [KEL00] os autores classificam dependências de acordo com diferentes características. Classificar as dependências permite tratá-las de uma forma sistemática e a utilização de uma metodologia de representação dos componentes e das dependências de um sistema, dentro de um escopo definido, possibilita a criação de ferramentas para melhorar tal sistema. Aumento da confiabilidade ou da escalabilidade são exemplo de melhorias que podem ser alcançadas.

Uma rede de dependências é composta por “*componentes*” interconectados por relações de dependência. Um componente tem um atributo de “**Tipo**”, definindo se é *software*, *hardware*, serviço, etc. Um componente tem ainda um atributo de “**Atividade**”, indicando se o mesmo é passivo ou ativo. Um componente ativo é aquele que dá início à uma ação, baseado em informações coletadas no sistema, em determinado horário ou mesmo a partir de uma ação gerada por outro componente. Um componente passivo é aquele que apenas reage a ações externas, não dando origem a novas ações.

A relação de dependência entre os componentes tem os seguintes atributos:

- **Domínio:** indica a distância entre componentes dependentes entre si; Esta distância varia em cada caso, por exemplo: componentes que compartilham a mesma memória, ou a mesma máquina, ou a mesma rede, etc, dependendo do modelo. Esta característica é ainda citada como *inter* vs. *intra* domínio;
- **Força da dependência:** quão forte é a dependência entre os componentes. Algumas vezes as dependências são binárias, ou seja, para que um componente simplesmente funcione, ele depende do funcionamento de outro componente. Outras vezes as dependências se apresentam em vários níveis de gradação, por exemplo, quando o desempenho de um componente afeta o desempenho de outro;
- **Formalização da dependência:** grau de formalização da dependência. Indica também se ela pode ser detectada automaticamente. Serve como uma métrica que ajuda a dimensionar o custo de determinar esta dependência, ou seu peso;
- **Criticidade da dependência:** indica como essa dependência deve ser satisfeita em termos da disponibilidade dos serviços, sistemas, etc. Por exemplo: pré-requisito, co-requisito, etc.

À parte destas características, é importante ainda considerar outro aspecto, o **tempo**, que determina o comportamento de dependências, caracterizando dependências dinâmicas. Uma relação de dependência pode existir em um momento e não existir algum tempo depois. É o caso, por exemplo, da necessidade de espaço em disco apenas durante a fase de instalação de um software.

2.2.Representação de dependências

É necessário representar o relacionamento entre componentes e, conseqüentemente, suas dependências, de maneira apropriada. Dois modelos se mostram adequados para representar tais relacionamentos: modelos *funcional* e *estrutural*. [ENS99] se refere a esses modelos como *abstract* e *environmental*, respectivamente. Nesse trabalho o autor define ainda outros modelos e propõe seu próprio, mas não trataremos destes neste texto.

Um modelo funcional representa o relacionamento genérico e independente de implementação ou instalação de serviços em termos das funcionalidades oferecidas e de que outros serviços estes dependem. Geralmente esses modelos são gerados manualmente por empresas fornecedoras de soluções ou equipamentos.

Um modelo estrutural por sua vez representa relacionamentos específicos e dependentes de implementação ou instalação e representa componentes reais do ambiente da organização. Este tipo de modelo é mais eficientemente gerado por processos automáticos, a partir de repositórios de pacotes instalados, por exemplo.

Assim, ao contrário do modelo estrutural que lida com componentes reais, o modelo funcional lida com elementos abstratos – classes com uma abstração do ambiente real.

A ferramenta mais adequada para se representar essas redes, dado seu poder de representação e operações, é o grafo dirigido. Neste trabalho consideraremos apenas grafos dirigidos acíclicos.

As figuras 1 e 2 ilustram ambos os modelos de dependência para um ambiente WEB simplificado. No primeiro caso temos as relações de dependência entre as funções abstratas, representadas por caixas na figura. As setas indicam relações de dependência diretas. Por exemplo, a função “*Browser WEB*” depende, para seu funcionamento, das funções “*Servidor WEB*”, “*Infra-estrutura de Rede*” e “*Servidor DNS*”.

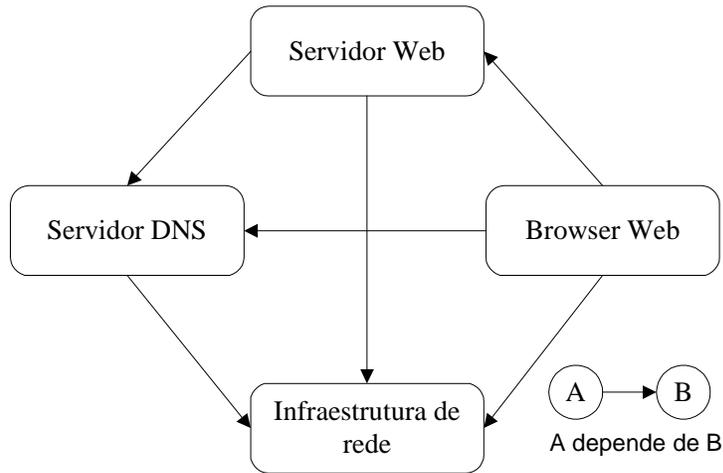


Figura 1: Modelo WEB funcional

Já no segundo caso, mostrado na figura 2, o modelo estrutural representa módulos instanciados para execução do ambiente WEB. Observa-se que existem dois “*Browser Web*” instanciados, cada um executando em uma máquina distinta.

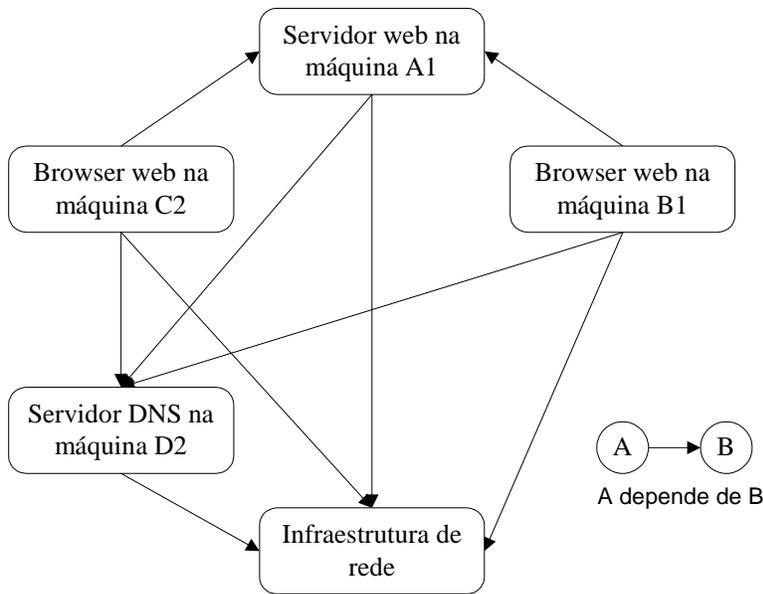


Figura 2: Modelo WEB estrutural

Os dois modelos tem utilidade no diagnóstico de falhas, sendo que o funcional possibilita um entendimento inicial e de nível mais alto da situação. O modelo estrutural, instanciado, possibilita encontrar os elementos específicos envolvidos nos problemas.

2.3. Um Método para geração de procedimentos para análise de falhas

Um modelo de dependência estrutural representando a rede de dependências real de uma extensa planta de equipamentos pode ser uma ferramenta bastante útil para isolar os componentes envolvidos em determinada tarefa no sistema. Entretanto, para dimensões amplas dos sistemas, esse modelo é de difícil concepção. Para o usuário, um modelo completo se mostra ainda de difícil leitura. Além disso existem freqüentes alterações na configuração do sistema, por exemplo, trocando-se *agentes* de máquinas, acrescentando *agentes* ou alterando a forma de comunicação com elementos de rede.

Uma ferramenta bastante útil para o operador, ao perceber uma falha em um dos módulos do sistema, é um roteiro passo-a-passo com as ações a serem tomadas para se testar sistematicamente cada módulo do sistema e serviços que podem causar a falha. Utilizando os modelos de dependência aqui mostrados, pode-se criar automaticamente um roteiro de verificação (*checklist*) do sistema, específico para uma falha ocorrida.

O método possui duas fases. A *Fase I* consiste em um preparo do sistema, através de dois passos feitos *a priori* e apenas uma vez, gerando as informações básicas para a fase seguinte. Os próximos passos, correspondentes à *Fase II*, são realizados dinamicamente, disparados sob demanda de um operador do sistema, para cada alarme desejado. A seguir são apresentados estes passos.

O primeiro passo é a construção do modelo funcional de dependências, contendo os módulos do sistema, os serviços e suas relações de dependências para o sistema. A construção deste modelo pode ser realizada manualmente, levando em consideração a experiência dos desenvolvedores e administradores do sistema, ou automaticamente. Neste trabalho consideramos a geração manual do modelo funcional, não são abordadas técnicas de geração automáticas para o modelo funcional.

O próximo passo é a identificação de ações para verificação do funcionamento de cada módulo. Por exemplo, ações para verificar se um *agente* está rodando, através do comando *ps*, ou a para verificação do funcionamento de determinada rede de dados através do uso do comando *ping*. Esta etapa também é feita manualmente e o conjunto de ações e comandos deve ser armazenada em uma base de dados ou dentro de uma aplicação apropriada.

Terminada a *Fase I*, o sistema tem as informações necessárias para realizar os passos da próxima fase para cada alarme desejado. Uma falha que ocorre em um elemento de rede é representada, para efeito de comunicação e armazenamento, em uma estrutura de dados denominada alarme. Quando uma falha ocorre em determinado ponto do sistema, mecanismos de auto-supervisão do sistema geram alarmes relatando problemas no funcionamento. Um operador, através de um *terminal* de operação, solicita então a geração do roteiro de *checklist* para a falha, disparando os passos da *Fase II*.

O terceiro passo então é, a partir do modelo funcional criado, gerar automaticamente o sub-grafo correspondente ao modelo estrutural específico, apenas com os componentes

envolvidos. Neste passo é usado um algoritmo de busca em largura no grafo correspondente à rede de dependências funcional, criado na *Fase I*, e é utilizado também a configuração do sistema, presente em um banco de dados de configuração. A cada passo do algoritmo de busca em que um novo componente é alcançado, os componentes daquele tipo que estiverem cadastrados no banco de dados, são adicionados ao modelo estrutural gerado, e as relações de dependência são adicionadas entre os nós do grafo.

Uma vez gerado esse modelo estrutural, um roteiro de análise de falhas será traçado e poderá ser utilizado na situação em questão, para a manutenção do sistema. O quarto passo então é a geração deste roteiro, realizada através de uma busca em largura no grafo de dependências estrutural gerado no passo anterior. A cada passo do algoritmo, ao se alcançar um nó no grafo, acrescentam-se as ações de diagnóstico previamente configuradas para aquele tipo de componente.

A geração do roteiro apresenta um checklist passo-a-passo de componentes, no caminho crítico da geração de falhas para um alarme específico percebido pelo operador do sistema. Desta forma, o operador saberá certamente se a solução do problema está sob sua alçada, ou se o mesmo deverá ser passado a outra equipe. Outro benefício é que no caso de acionamento de equipes, por exemplo através da abertura de um bilhete de reparo, o operador já poderá passar os dados coletados, como ping's que não funcionaram ou excesso de consumo de CPU em determinada máquina.

3. O SIS como um sistema distribuído de gerenciamento

Para estudar e entender as relações de dependência entre módulos e entre processos computacionais em sistemas distribuídos, bem como aplicar os conceitos até então desenvolvidos, usamos como objeto de trabalho o SIS, Sistema Integrado de Supervisão, um sistema distribuído de gerenciamento de redes de computadores e de telecomunicações, responsável por diversas atividades de gerenciamento, tais como gerenciamento de falhas, provisionamento, coleta de dados de tráfego e tarifação de centrais, entre outros.

A escolha deste sistema se deve às suas características bastante propícias para este trabalho, uma vez que o SIS é um sistema distribuído com módulos de alto grau de interdependência que se comunicam basicamente utilizando protocolos de transporte e rede (RPC e TCP/IP). A distribuição dos módulos comunicantes é muito variável, desde a comunicação entre processos em uma mesma máquina, passando pela comunicação em rede local, até a comunicação remota, em redes de longa distância - WAN.

Dada a enorme área geográfica de abrangência do sistema, atualmente em 16 estados da federação, e o grande número de equipamentos e módulos envolvidos, a complexidade de gerenciamento do próprio sistema de gerenciamento tende a crescer. Consequentemente, problemas relacionados ao desempenho do sistema e sua escalabilidade se mostram algumas vezes de difícil análise e solução. Desta forma, a necessidade de tratamento desses problemas se tornam uma das motivações deste trabalho.

A figura 3 mostra a estrutura do SIS em uma região de operação típica. O sistema é organizado hierarquicamente, de forma a melhor servir a uma grande área geográfica, ou a uma área de grande concentração de equipamentos, com suas demandas de desempenho. Por

exemplo, uma grande capital, em média, é dividida no sistema em quatro sub-regiões com estruturas como a apresentada na figura 3, que estão sob a supervisão de uma região.

Como o sistema é flexível, pode-se configurar um SGBD (Sistema Gerenciador de Banco de Dados) para uma sub-região, região ou para uma área maior, dependendo do tamanho, concentração de equipamentos e desempenho desejados. Usualmente utiliza-se um SGBD para cada região.

No exemplo, uma sub-região tem sete gerentes, e em média cada gerente se comunica com treze agentes. Na maioria das vezes cada agente supervisiona um único elemento de rede, mas alguns agentes supervisionam vários elementos de rede, como o A13 da figura 3, que supervisiona 2 elementos de rede.

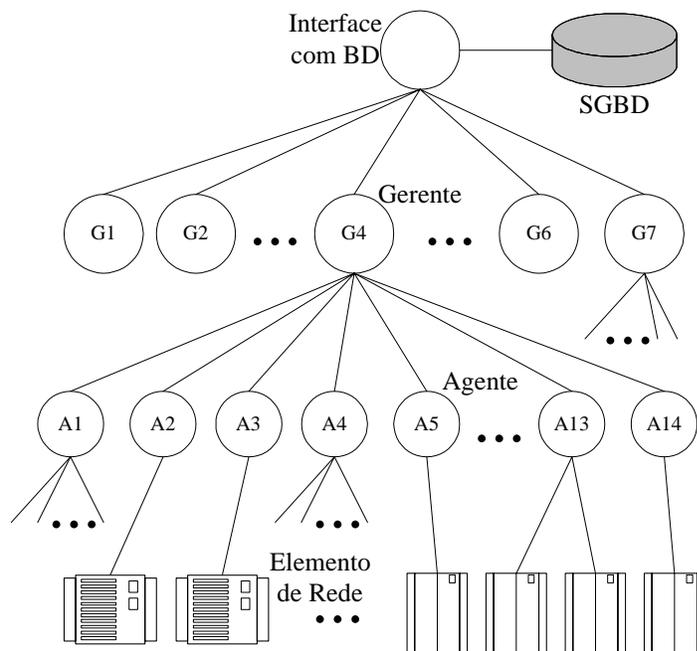


Figura 3: Estrutura típica do SIS

Dentre as diversas tarefas realizadas pelo SIS, a pioneira, e uma das principais, é o gerenciamento de falhas, que consiste na coleta de falhas dos equipamentos gerenciados e sua disponibilização para consultas, através de seu armazenamento em um **banco de dados de falhas**. Os módulos e equipamentos envolvidos na realização desta tarefa fazem parte do caminho crítico do gerenciamento de falhas do SIS.

Outra tarefa, igualmente importante, é a configuração automática de terminais de assinantes. Esta tarefa contém módulos em comum com a tarefa de gerenciamento de falhas, bem como alguns módulos distintos. Assim tais módulos formam o caminho crítico do gerenciamento de configuração do SIS.

Serão apresentados a seguir maiores detalhes sobre gerenciamento de falhas, sem perda de representatividade para gerenciamento de configuração e outras áreas de gerenciamento do sistema, uma vez que os conceitos são similares.

3.1. Coleta e armazenamento de alarmes

Uma falha que ocorre em um elemento de rede é representada, para efeito de comunicação e armazenamento, em uma estrutura de dados denominada alarme.

No caminho compreendido entre a ocorrência de uma falha em um determinado elemento de rede (no contexto do SIS, entidade supervisionada - ES) e a consulta dos alarmes por parte dos operadores, existe uma variedade de módulos de software, processos e infra-estrutura envolvidos. Os módulos encontram-se num ambiente distribuído e apresentam relacionamentos e dependências entre si, além da infra-estrutura da qual são dependentes para a realização de suas tarefas, como rede de computadores, máquinas, energia, etc.

A figura 4 mostra os módulos do SIS envolvidos no caminho de coleta e armazenamento de alarmes, no exemplo, de uma *central telefônica*. Os componentes e arestas mostram alguns potenciais pontos de dependência no processo de coleta da falha, comunicação entre os módulos e armazenamento no banco de dados. As linhas indicam que existem comunicação entre os módulos conectados. Na próxima seção veremos o modelo funcional para a coleta de falhas no SIS, e quais componentes e dependências serão analisados no modelo.

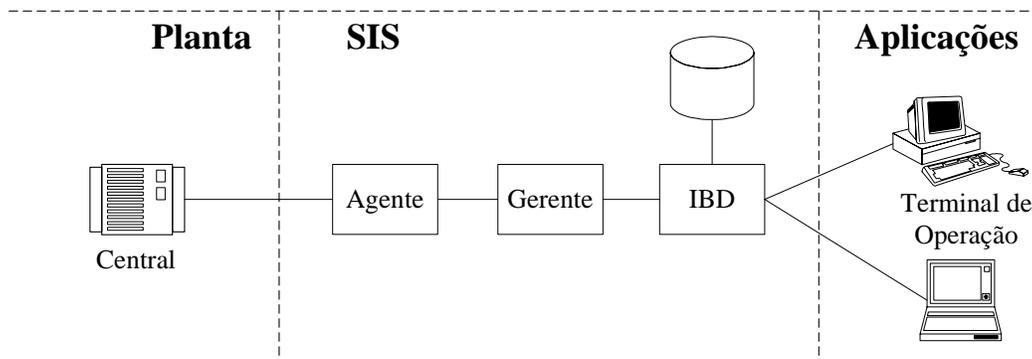


Figura 4: Caminho crítico da coleta de falhas

O módulo *agente* depende da correta comunicação com a central para que ele possa interrogá-la em busca de alarmes notificando falhas. Por outro lado o agente também depende do módulo *gerente* para solicitar o envio dos alarmes coletados para armazenamento. O gerente por sua vez depende do módulo de interface com o banco de dados (*IBD*) para repassar os alarmes para o banco de dados. O **terminal de operação** do usuário também depende do módulo interface com o banco de dados para obter os dados de suas consultas e realizar as atualizações necessárias.

4. Aplicando o Método para Análise de Falhas no SIS

Este trabalho se propõe a montar uma rede de dependências entre os diversos módulos de software para o caminho citado acima, através de um modelo funcional. Apesar da gama de características propostas pelos autores em [KAR00] representadas na seção 2.1, neste trabalho vamos considerar apenas a característica *tipo* de componente. O *peso* das dependências será considerado binário, ou seja, se existe uma dependência entre dois componentes significa que apenas o funcionamento do sistema ou serviço, e não seu desempenho, está representado por esta dependência.

Apresentaremos um modelo funcional de dependências de um sistema distribuído de gerenciamento de redes de telecomunicações, e através deste modelo e da configuração do sistema, presente em um banco de dados de configuração, faremos a geração do modelo estrutural. A partir do modelo estrutural, pode-se criar roteiros para resolução de problemas (*troubleshooting*) para os técnicos de plantão e administradores do sistema, assim como outros usos como correlação de alarmes, correção automática, etc.

Para a criação do modelo no sistema proposto, serão considerados os serviços e módulos representados pelos seguintes componentes:

- **Elemento de Rede – ER:** Elemento da planta gerenciado. Por exemplo, centrais de comutação;
- **Enlace com ER:** Serviço de conexão com o ER. Atualmente é possível a conexão de centrais telefônicas através de várias formas, tais como TCP/IP, Rede de pacotes X.25, linhas dedicadas (LPCD's), protocolos proprietários, etc;
- **Agente:** Módulo de software do SIS para comunicação e supervisão de ER. Um agente é específico para a tecnologia de equipamento que supervisiona;
- **Gerente:** Módulo de software do SIS para comunicação e supervisão dos agentes. Todos os gerentes são idênticos e sua forma de comunicação com os agentes faz parte de um protocolo aberto do SIS;
- **Interface BD – IBD:** Módulo de software do SIS de acesso ao SGBD. Todas as consultas e atualizações são realizadas através deste módulo; que implementa uma API;
- **Rede Local – LAN:** Serviço de rede local de interligação de máquinas e equipamentos, atualmente predominantemente da tecnologia TCP/IP;
- **Rede WAN:** Serviço de rede de longa distância interligando outras LAN's.

Outros recursos ou serviços são também necessários para o correto funcionamento do SIS, tais como a máquina hospedeiro em que cada módulo executa, sistema operacional, serviços de NIS ou DNS para resolução de nomes, entre outros. Entretanto não trataremos dessas dependências explicitamente neste trabalho.

A figura 5 mostra a modelagem funcional para a atividade de coleta de dados de falhas do SIS. Nesta atividade, um agente depende do correto funcionamento dos elementos de rede nas suas funções de comunicação e de gerenciamento. Além disso, depende do serviço de comunicação com as centrais. Todos os módulos do SIS neste caminho dependem de suas redes locais (LAN), assim como do correto funcionamento do serviço de ligação das redes (WAN).

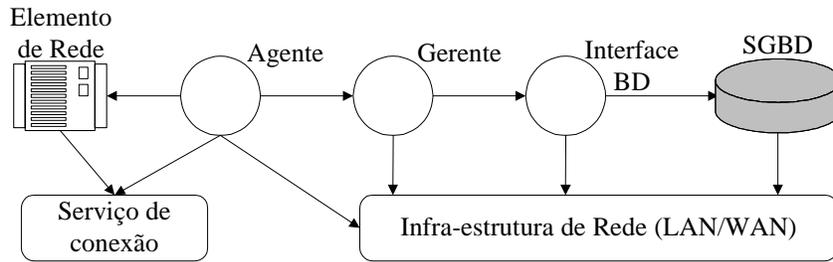


Figura 5: Modelo funcional de dependências da coleta de falhas no SIS

Já a figura 6 apresenta um modelo estrutural de dependências para a coleta de dados. O modelo foi simplificado para fins de compreensão, sem perda de representatividade. Por exemplo, um gerente comunica-se com cerca de 15 agentes tipicamente. Além disso, muitas vezes dezenas de redes locais e de longa distância diferentes são utilizadas pelos módulos do SIS e pelos elementos de rede.

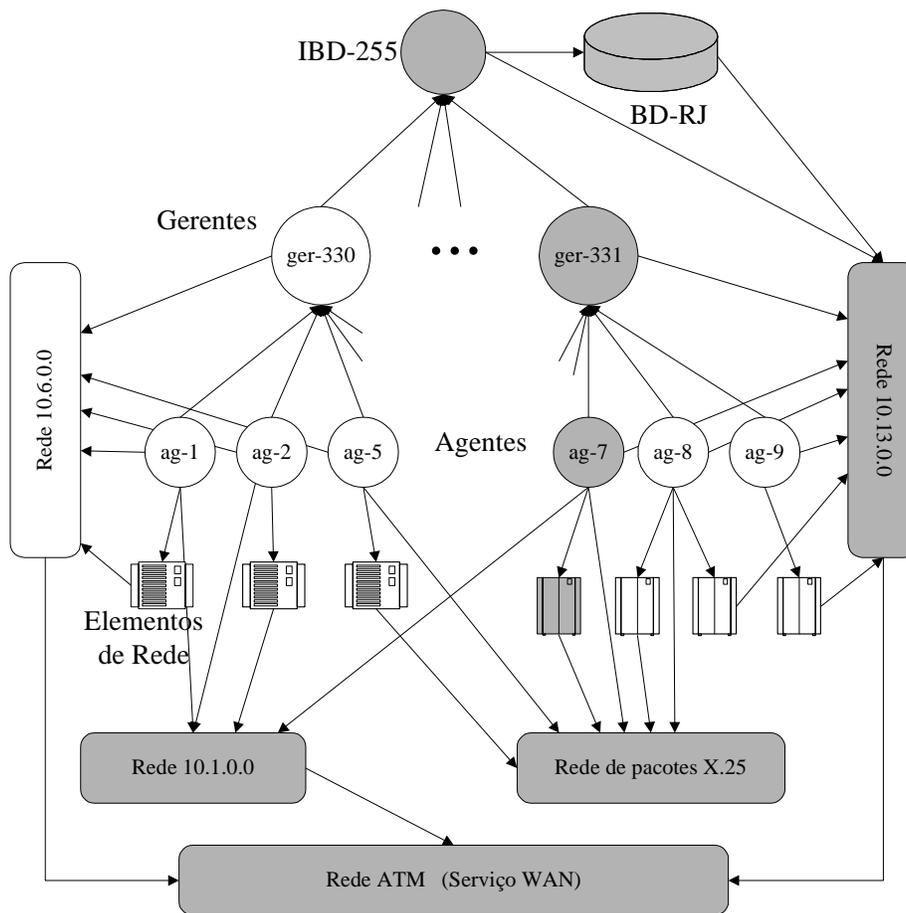


Figura 6: Modelo estrutural de dependências da coleta de falhas no SIS

Com essa figura pode-se ter uma idéia mais realista da complexidade do relacionamento entre os módulos do sistema e dos recursos empregados para a coleta de falhas de um único elemento de rede. Por exemplo, os componentes em realce na figura

representam os recursos envolvidos na coleta das falhas do elemento de rede, no exemplo uma central telefônica, supervisionada pelo agente *ag-7*. Este agente roda em um hospedeiro da rede 10.1.0.0, e depende naturalmente desta rede. O elemento de rede que o mesmo gerencia utiliza a rede de pacotes para se ligar ao sistema; assim, esta rede também é dependência para este agente. Já o gerente *ger-331* roda em um hospedeiro da rede 10.13.0.0 e o agente em questão necessita se comunicar com ele, para repassar as falhas coletadas da central. As duas LAN's envolvidas precisam ser conectadas, e assim dependem do serviço WAN, implementado por rede ATM 155 Mbps.

Todos esses recursos utilizados pelo sistema representam uma fonte potencial de falhas e problemas. Além disso, muitas vezes, diferentes recursos são administrados por equipes diferentes, seja por um motivo geográfico de administração de pessoal (exemplo de redes locais), ou por causa de suas características distintas (equipes de SGBD's, equipes de comunicação de dados, etc). Desta forma, ao ocorrer uma falha em um componente do sistema ou em um recurso utilizado, é importante identificar apropriadamente o *ofensor* do problema, ou seja, a causa principal do mesmo. Quando a causa principal é bem definida, é possível acionar a equipe apropriada para tratá-la, reduzindo o risco de trabalho desnecessário, acionamentos indevidos (para o caso de equipes de plantão) e atraso na resolução do problema.

4.1. Geração de procedimentos de análise de falhas

Abaixo é apresentada a aplicação do método aqui desenvolvido na seção 2 para o SIS. A *Fase I* foi realizada inicialmente, com a construção manual do modelo funcional, apresentado na figura 5 e o levantamento de comandos para os testes de software e rede.

A seguir foi escolhido um alarme “AGENTE NÃO RESPONDE” para o agente *ag-7*, representado na figura 6, e executado o terceiro passo, a geração do modelo estrutural para este módulo. A figura 7 mostra o sub-grafo gerado para o agente escolhido. Essa figura corresponde aos módulos realçados no modelo representado na figura 6 e suas dependências.

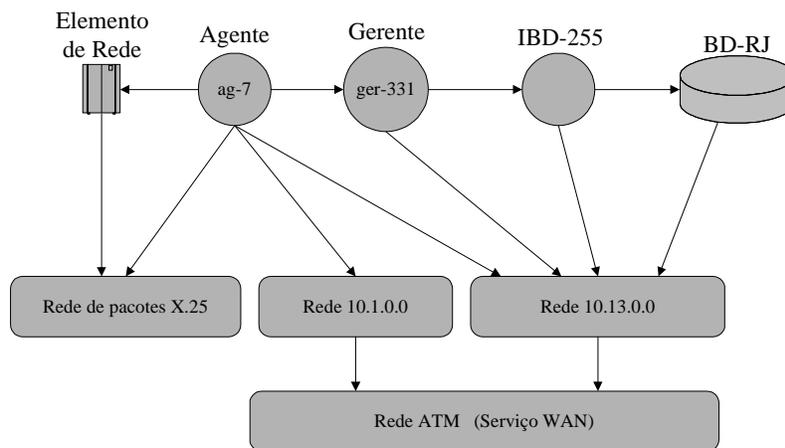


Figura 7: Modelo estrutural de dependências para o agente *ag-7*

Para a execução deste passo, por exemplo a partir de um “Agente” no modelo funcional representado na figura 5, o algoritmo realiza a pesquisa encontrando o nodo “Elemento de Rede”. No modelo estrutural correspondente, considerando o *ag-7* na figura 6, será incluído o elemento de rede associado. Em seguida o algoritmo encontrará “Gerente” e *ger-331* será incluído ao modelo estrutural, e assim por diante, até a geração do modelo mostrado na figura 7.

O próximo passo corresponde à geração do roteiro de análise de falhas. Este passo é executado através da busca em largura no modelo estrutural gerado no passo anterior, como descrito na seção anterior. Abaixo é mostrado um exemplo de roteiro de testes para o alarme de “AGENTE NÃO RESPONDE”, para o agente *ag-7*. Em cada passo do processo o usuário estará realizando um teste específico em cada recurso, e em caso de falha em determinado passo, o usuário deverá identificar a equipe responsável pela normalização do problema.

1. Entrar na máquina hospedeiro do agente *ag-7* (10.1.0.12) com o usuário **sis**.
2. Usar o comando **ping** para verificar a comunicação desta máquina com a máquina hospedeiro do gerente *ger-331* (10.13.0.4).
3. Verificar se o processo do agente *ag-7* está rodando através do comando **ps**.
4. Verificar o uso de CPU e memória na máquina através dos comandos **ps** e **top**.
5. Entrar na máquina hospedeiro do gerente *ger-331* (10.13.0.4) com o usuário **sis**.
6. Verificar se o processo do gerente *ger-331* está rodando através do comando **ps**.
7. Verificar se o processo da interface BD *intbd-255* está rodando através do comando **ps**.
8. Verificar o uso de CPU e memória na máquina através dos comandos **ps** e **top**.
9. Usar o comando **ping** para verificar a comunicação desta máquina com a máquina hospedeiro do SGBD *BD-RJ* (10.13.0.1).
10. Entrar na máquina hospedeiro do SGBD *BD-RJ* (10.13.0.1) com o usuário **sis**.
11. Verificar se o processo do SGBD *BD-RJ* está rodando através do comando **ps**.
12. Verificar o uso de CPU e memória na máquina através dos comandos **ps** e **top**.

Outros usos para os modelos apresentados neste trabalho poderão ser explorados futuramente. Uma possibilidade que está sendo explorada é a realização automática dos testes descritos no roteiro acima. Uma vez determinadas as dependências e instrumentado o sistema, limites de operação do mesmo poderão ser determinados, bem como possíveis gargalos serão mostrados. O resultado desta tarefa poderá levar ao estudo de formas de aumentar a escalabilidade do sistema, ou de sua confiabilidade e tolerância a falhas.

5. Conclusões

Os sistemas distribuídos de gerenciamento para redes de telecomunicações caracterizam-se por uma ampla distribuição geográfica, grande número de componentes com alta relação de dependência. Similarmente a outros sistemas, estão sujeitos a falhas de funcionamento de seus módulos, que podem levar a situações de difícil diagnóstico. Neste trabalho apresentamos o resultado de uma pesquisa e desenvolvimento que aplicam técnicas de redes de dependência para auxiliar no diagnóstico de falhas em tais sistemas.

Usando como ambiente de experimentação o SIS verificamos a viabilidade da construção de mecanismos com um elevado grau de automatização para os propósitos de detecção de falhas e auxílio na identificação de causa principal.

As representações de redes de dependência apresentadas neste trabalho servem como ferramenta básica para outros trabalhos que podem ser executados, como correlação de alarmes, planejamento de capacidade, testes automatizados, entre outros.

A técnica de geração de roteiros de diagnósticos apresentada pode ser expandida para contemplar testes mais elaborados ou complexos, incluir outros componentes ou mesmo ser parcialmente automatizada. Este último caso encontra-se atualmente em desenvolvimento.

6. Referências Bibliográficas

- [NOG96] J. M. S. Nogueira, D. M. Meira. “*The SIS Project: A Distributed Platform for the Integration of Telecommunication Management System*”. In proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS), Kyoto, Japan, 1996.
- [YEM96] S. Yemini, S. Kliger et al. “*High Speed and Robust Event Correlation*”. IEEE Communications Magazine 34(5):82-90, Maio de 1996.
- [CHO99] J. Choi, M. Choi and S. Lee. “*An alarm correlation and fault identification scheme based on OSI management object classes.*” 1999 IEEE International Conference on Communications, 1547-51, Vancouver, BC, Canadá, Junho de 1999.
- [SAF99] G. P. Safe, J. M. Nogueira e C. Machado. “*An automatic fault diagnosis and correction system for telecommunications management*”. Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM’99), Maio, 1999, Boston, MA, USA.
- [GRU98] B. Grusche. “*Integrated Event Management: Correlation Using Dependency Graphs*”. In Proceedings of 9th IFIP/IEEE International Workshop on Distributed Systems Operation & Management (DSOM’98), 1998.
- [MEI00] D. M. Meira, and J. M. S. Nogueira. “*A recursive approach for alarm correlation in telecommunication networks*”. IEEE/IFIP Network Operations and Management Symposium, 2000.

- [MEI97] D. M. Meira. “*Um modelo para correlação de alarmes em redes de telecomunicações*”. Tese de doutorado defendida no DCC/UFMG. Novembro de 1997.
- [KAT97] S. Katker and M. Paterok. “*Fault isolation and event correlation for integrated fault management* ”. In Proceedings of the fifth IFIP/IEEE International Symposium on Integrated Network Management (IM V), 583-596, San Diego, CA, Maio de 1997.
- [ENS99] C. Ensel. “*Automated generation of dependency models for service management*”. Workshop of the OpenView University Association (OVUA'99), Bologna, Itália, Junho de 1999.
- [KEL00] A. Keller, U. Blumenthal and G. Kar. “*Classification and Computation of Dependencies for Distributed Management*”, In Proceedings of the fifth IEEE Symposium on Computers and Communications (ISCC 2000), Antibes, Juan-les-Pins, França, Julho de 2000.
- [KAR00] G. Kar, A. Keller, and S.Calo. “*Managing application services over service provider networks: Architecture and dependency analysis*”, In Proceedings of the seventh IFIP/IEEE Network Operations and Management Symposium (NOMS 2000), Honolulu, HI, Abril de 2000.
- [BRO01] A. Brown, A. Keller, G. Kar. “*An active approach to Characterizing Dynamic Dependencies for Problem Determination in a Distributed Application Environment* ”. IFIP/IEEE International Symposium on Integrated Network Management (IM'2001), Seattle, WA, Maio 2001.
- [THO01] D. Thoenen, J. Riosa, J. L. Hellerstein. “*Event Relationship Networks: A Framework for Action Oriented Analysis in Event Management*”. Artigo a ser apresentado no IFIP/IEEE International Symposium on Integrated Network Management (IM'2001), Seattle, WA, Maio 2001.