

6

Considerações Finais e Trabalhos Futuros

A sociedade moderna tem se tornado cada vez mais dependente de serviços de telecomunicações como telefonia e acesso à Internet. Tais serviços são disponibilizados através de redes de telecomunicações cada vez maiores e mais complexas, e que são compostas por equipamentos de diferentes tipos e tecnologias. Para administrar tais redes de forma eficiente e economicamente competitiva, diferentes modelos e estratégias de gerência de redes têm sido desenvolvidos e empregados ao longo do tempo. Cada novo modelo é construído com base nos sucessos e fracassos dos modelos anteriores, num campo de pesquisa que tem se mostrado bastante dinâmico.

Este trabalho explorou alguns dos principais conceitos da área de gerência de redes, apresentando os mais importantes e tradicionais padrões de gerência, assim como novas tecnologias que têm sucedido estes padrões e que são baseadas em abstrações tais como objetos distribuídos e código móvel, por exemplo. Uma ênfase especial foi dada ao uso de agentes de software aplicados ao domínio de gerência de redes. A utilização de agentes de software é bastante apropriada neste contexto em que pró-atividade, autonomia e flexibilidade são buscados com tamanha propriedade.

Neste trabalho foi apresentada uma arquitetura baseada em agentes de software para a gerência de falhas em redes legadas de telecomunicações. Foram também exibidos dois estudos de caso que ilustram como tal arquitetura pode ser utilizada para a modelagem de testes. O objetivo principal da arquitetura é o de possibilitar o monitoramento e a aplicação de rotinas para o diagnóstico e a correção de falhas em redes gerenciadas por sistemas centralizados. Destaque deve ser dado para o fato de que tais testes pode ser feitos de forma distribuída. Pode-se considerar que este objetivo faz parte de uma meta de escopo mais abrangente que é a de possibilitar que tais redes legadas satisfaçam novos e mais exigentes requisitos de operação. Como foi discutido no decorrer do trabalho, esta meta é relevante para muitas organizações pois tais redes podem continuar em operação por muitos anos mesmo depois de serem consideradas antiquadas e ultrapassadas por soluções baseadas em tecnologias mais modernas. Não é uma tarefa árdua imaginar-se um exemplo de como tal arquitetura pode ser útil. Um sistema com

testes eficazes para a detecção de falhas construído sobre a arquitetura proposta pode diagnosticar falhas rapidamente, ou mesmo previni-las, contribuindo para um menor tempo de indisponibilidade da rede e, conseqüentemente, alcançando um melhor nível de qualidade do serviço exigido por clientes ou imposto pelo mercado. Caso a permanência em operação de tais redes legadas seja de fato necessária, soluções como a aqui apresentada podem contribuir no sentido de aumentar sua vida útil. Isto se torna possível na medida em que novos requisitos operacionais podem ser satisfeitos, e a defasagem em relação a redes mais modernas pode ser diminuída.

Naturalmente, para que tais objetivos sejam alcançados com êxito, algumas restrições devem ser observadas. Uma das mais importantes é que sistemas baseados na arquitetura proposta devem exercer o mínimo de carga sobre o sistema de gerência existente, de modo a não prejudicar o seu funcionamento regular. Para tanto, foi proposto um mecanismo que distribui informações de topologia e inventário por agentes de software distribuídos pela rede. Isto reduz a freqüência necessária de interação com o sistema de gerência e, conseqüentemente, proporciona um menor impacto sobre o mesmo. Para tanto, a rede é dividida em partições, definidas por critérios geográficos. Para cada partição existe um agente responsável por fornecer dados de topologia para os demais agentes do sistema.

Outra importante contribuição deste trabalho é a modelagem de ações de diagnóstico de falhas através de testes e tarefas. O uso destas abstrações, em particular a das tarefas, permite uma maior reutilização de código em sistemas de diagnósticos de falhas em redes de telecomunicações. Isto é possível uma vez que as comunicações com equipamentos de rede são isoladas em elementos atômicos com interfaces de entrada e saída bem definidas. Assim sendo, cada tarefa pode ser facilmente utilizada em várias situações, o que pode diminuir os esforços de criação de testes novos. Além do benefício da reutilização de código, outra importante característica desta modelagem baseada em testes e tarefas é a sua flexibilidade no que diz respeito ao tipo de ações que podem ser implementados através dela. Como há poucas restrições em relação à implementação das tarefas, praticamente qualquer tipo de comunicação com equipamentos pode ser desempenhado. Não há restrição, por exemplo, ao tipo de protocolo de gerência utilizado para tal comunicação. Pode-se utilizar protocolos como SNMP e TL1, ou mesmo CLI. Quando da implementação de um teste, além da flexibilidade obtida através das tarefas, há ainda a possibilidade da obtenção de informações de topologia, o que expande ainda mais o universo de testes que este modelo é capaz de contemplar.

Por fim, uma última questão é que, apesar de este sistema ter sido concebido com o foco na área de gerência de falhas, também é possível utilizá-lo sem praticamente nenhuma alteração em outras áreas de gerência de redes, como a

gerência de configuração, por exemplo. Tal característica se deve à flexibilidade obtida através dos testes e tarefas, que podem ser adaptados para ações de gerência e não apenas diagnóstico e correção de falhas.

Apesar de a arquitetura aqui proposta atender aos objetivos inicialmente estabelecidos, há ainda bastante espaço para melhorias da mesma. Algumas destas melhorias são identificadas e discutidas a seguir.

A questão possivelmente mais importante que não foi considerada no escopo deste trabalho foi a criação e a aplicação de uma ontologia para modelagem dos recursos de rede e dos testes que podem ser efetuados sobre a mesma. Esta ontologia seria uma peça fundamental para possibilitar interações mais ricas do ponto de vista semântico entre os agentes de software. Parte deste problema foi solucionada neste trabalho com o uso de esquemas XML para modelagem dos recursos de rede. Pouco foi feito, entretanto, para modelar os testes de forma que os agentes pudessem ponderar ou deliberar sobre os mesmos. O uso de ontologias seria uma condição necessária para que os agentes de software, que neste trabalho são fundamentalmente reativos, pudessem assumir também características deliberativas. Um bom ponto de partida para esta questão de ontologias de gerência de redes pode ser encontrado em [Lavinal06].

Outra área em que há espaço para avanços é a criação de mecanismos para que o local de execução de um teste possa migrar. Isto seria útil em testes em que a comunicação com equipamentos localizados em partições diferentes da rede é necessária. No estado atual, limitações da arquitetura exigem que toda execução de um teste seja conduzida a partir de um ponto único. Este ponto é necessariamente um agente de teste, que é imóvel. Uma alternativa para a resolução deste problema seria permitir que os agentes de testes fossem móveis, deslocando-se pela rede. Outra alternativa seria criar um mecanismo para a transferência da execução de testes entre agentes de testes.

Outro aspecto em que há espaço para melhorias diz respeito ao fato de que os agentes de testes modelados neste trabalho estão preparados para coletar informações de equipamentos e sistemas de gerência, mas não para receber alarmes e notificações dos mesmos. Há espaço aqui para a criação de um novo tipo de agente de software que possua mecanismos para monitorar alarmes de equipamentos, *logs* de sistemas de gerência e similares e que tome as devidas ações em função destas informações coletadas. Esta estratégia ajudaria a introduzir mais autonomia na arquitetura proposta.

Por fim, uma última frente para novos desenvolvimentos seria a pesquisa e aplicação de algoritmos e estratégias para previsão de falhas e a adaptação dos mesmos no contexto da arquitetura proposta neste trabalho.