

Gestão de Redes Baseada em Agentes Móveis

Ricardo Jorge Teixeira da Costa

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Arquiteturas, Sistemas e Redes**

Orientador: Doutora Maria João Monteiro Ferreira Viamonte

Co-orientador: Mestre Joaquim Ricardo Azevedo Teixeira

Júri:

Presidente:

Doutor Luis Miguel Moreira Lino Ferreira, Professor Adjunto, Instituto Superior de Engenharia do Porto

Vogais:

Doutora Berta Hermínia Paradinha Baptista Dias Pinheiro, Professor Adjunto, Instituto Superior de Engenharia do Porto

Doutora Maria João Monteiro Ferreira Viamonte, Professor Adjunto, Instituto Superior de Engenharia do Porto

Porto, Outubro de 2014

Aos meus pais

Resumo

A gestão das redes informáticas converteu-se num fator vital para as redes operarem de forma eficiente, produtiva e lucrativa. A gestão envolve a monitorização e o controlo dos sistemas para que estes funcionem como o pretendido. Ações de configuração, monitorização e reconfiguração dos componentes, são essenciais para melhorar o desempenho, diminuir o tempo de inatividade, melhorar a segurança e efetuar contabilização. A maioria das aplicações de gestão de Redes assenta no paradigma centralizado cliente-servidor, onde um servidor central coleta e analisa dados provenientes dos diferentes elementos dispersos pela rede. Esses dados de gestão estão armazenados em bases de dados de gestão localizadas nos diversos elementos da rede. No entanto, este paradigma não tem conseguido dar resposta às exigências das redes atuais.

Surge assim a necessidade de utilizar novos paradigmas para a gestão das redes. Uma alternativa baseada no paradigma dos agentes móveis já foi estudada, proposta e desenvolvida para a gestão da rede do GECAD-ISEP. Neste trabalho pretende-se propor e adicionar novas capacidades aos agentes móveis existentes, assim como novos agentes. Com vista à validação da solução proposta recorre-se à utilização de um simulador de redes.

Palavras-chave: Gestão de Redes, *Simple Network Management Protocol*, Elementos de Rede, Agentes Móveis, Simulador de Redes.

Abstract

The management of computer networks has become a vital networks operate efficiently, productively and profitably factor. The management involves monitoring and control systems so that they function as intended. Configuration actions, monitoring and reconfiguration of components, are essential to improve performance, reduce downtime, improved safety and making accounting. Most networks management applications based on the centralized client-server paradigm, where a central server collects and analyzes data from different elements dispersed over the network. These management data are stored in databases located in several managed elements of the network. However, this paradigm has failed to meet the requirements of today's networks.

This creates the need for new paradigms for network management. One based on the paradigm of mobile agents has been studied alternative, proposed and developed for the management of the GECAD-ISEP network. In this work we intend to propose and add new capabilities to existing mobile agents, and new agents. For validation of the proposed solution resorts to the use of a network simulator.

Keywords: Network Management, Simple Network Management Protocol, Network Elements, Mobile Agents, Network Simulators.

Agradecimentos

Gostaria de agradecer a todos os que, de uma forma direta ou indireta, contribuíram para a realização deste trabalho, no entanto, não posso deixar de salientar a grande contribuição dada pelos meus colegas Marco Silva, Gabriel Santos, Tiago Miguel e Tiago Sousa.

À minha orientadora, Doutora Maria João Monteiro Ferreira Viamonte e ao co-orientador Mestre Joaquim Ricardo Azevedo Teixeira, por toda a disponibilidade e ajuda, que foram fundamentais para a realização deste trabalho.

Ao Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão por todos os recursos disponibilizados para a realização deste trabalho.

Por fim, a toda a minha família e amigos, pelos conselhos dados e apoio manifestado continuamente. Igualmente, não poderia deixar de evidenciar o meu sincero reconhecimento e gratidão aos meus pais pelo enorme amparo prestado ao longo da minha vida, por estarem sempre presentes e pelo esforço que têm dedicado e incondicionalmente exibido, no decurso de todos estes anos, para que realize as minhas ambições e objetivos. Sem eles, não seria nada nem ninguém.

Índice

1	Introdução	1
1.1	Enquadramento	1
1.2	Objetivos e Principais Contribuições.....	2
1.3	Organização da Dissertação	3
2	Introdução à Gestão de Redes	5
2.1	Introdução e Definições	5
2.2	Importância da Gestão de Redes	7
2.3	Áreas Funcionais da Gestão de Redes	9
2.4	Modelo Arquitetural da Gestão de Redes	12
2.4.1	Arquiteturas Centralizadas	12
2.4.2	Arquiteturas Descentralizadas	13
2.5	Modelo de Informação de Gestão.....	16
2.6	Modelo Relacional de Gestão.....	19
2.6.1	SNMP.....	19
2.7	Conclusões	22
3	Gestão Baseada em Agentes	23
3.1	Definição de Agente	23
3.2	Tipos e Classificação de Agentes	23
3.3	Sistemas Multiagente	24
3.4	Agentes Móveis.....	29
3.4.1	Características dos Agentes Móveis	29
3.4.2	Arquitetura Cliente-Servidor vs Agentes Móveis	30
3.4.3	Vantagens na Utilização de Agentes Móveis	31
3.4.4	Conclusão	33
4	Simulador de Redes	35
4.1	Introdução	35
4.2	Ferramentas de Simulação - Principais características e funcionalidades	35
4.2.1	<i>Cisco Packet Tracer</i>	35

4.2.2	<i>Psimulator 2</i>	41
4.2.3	<i>Graphical Network Simulator - GNS 3</i>	44
4.3	Conclusão	48
5	Mobile Agent Network Management Application	51
5.1	Introdução	51
5.2	Arquitetura	52
5.2.1	Visão geral.....	52
5.2.2	Agentes Móveis da Comunidade	53
5.2.3	Interação entre os Agentes	55
5.2.4	Limitações da Aplicação baseada em Agentes	60
5.2.5	Proposta de melhoramento.....	61
6	Solução Proposta	63
6.1	Introdução	63
6.2	Detalhes de Implementação.....	63
6.2.1	Agente <i>AgentPrinter</i>	63
6.2.2	Novas Funcionalidades	66
6.3	Cenário no GNS 3	70
6.4	Resultados	72
7	Conclusões e Trabalho Futuro	81
7.1	Resumo	81
7.2	Objetivos Alcançados	82
7.3	Limitações e Trabalho Futuro	83

Lista de Figuras

Figura 1- Uma Rede e a sua Organização.....	6
Figura 2- O papel da Gestão de Redes	6
Figura 3- Componentes da Gestão de Redes.....	7
Figura 4- Gestão da Rede, Sistemas e Aplicações.....	7
Figura 5- As cinco áreas funcionais da gestão de redes do modelo OSI.....	10
Figura 6- Exemplo de Arquitetura Centralizada (baseado em [6])	12
Figura 7- Arquitetura Hierárquica (baseada em [6]).....	14
Figura 8- Arquitetura baseada em Agentes Móveis (baseada em [6])	15
Figura 9- Arquitetura altamente distribuída [38].....	15
Figura 10- MIB II [32].....	17
Figura 11- MIB Browser.....	18
Figura 12- Tabela resultante da operação "GET-BULK"	18
Figura 13 - Arquitetura do SNMP ([baseada em 36]).....	20
Figura 14- Entidade SNMP [32]	21
Figura 15 - Estrutura de um SMA (baseada em [42]).....	25
Figura 16 - Comunicação num Agente	27
Figura 17- Comunicação Direta (baseada em [42]).....	27
Figura 18- Comunicação Assistida (baseada em [42])	28
Figura 19- Arquitetura Cliente-Servidor.....	30
Figura 20- Arquitetura Agentes Móveis.....	30
Figura 21- Abordagem Cliente-Servidor vs Agentes Móveis.....	31
Figura 22- Execução Assíncrona	32
Figura 23 - Interface do simulador <i>Cisco Packet Tracer</i>	36

Figura 24- Dispositivos suportados pelo simulador	37
Figura 25- Exemplo de adição de módulos num <i>router</i>	37
Figura 26- Opções de Configuração	38
Figura 27- " <i>Tag</i> " <i>Desktop</i>	38
Figura 28- Tipos de conexão no <i>Packet Tracer</i>	39
Figura 29- Área Lógica vs Área Física.....	40
Figura 30- Protocolos suportados pelo <i>Packet Tracer</i>	41
Figura 31- Interface gráfica do <i>PSimulator</i>	41
Figura 32 - Exemplo de escolha de um dispositivo no <i>PSimulator</i>	42
Figura 33 - Janela de configuração de um <i>router</i> no <i>PSimulator</i>	43
Figura 34 - Análise do tráfego de pacotes no <i>PSimulator</i>	44
Figura 35 - Resultado de um <i>traceroute</i> no <i>PSimulator</i>	44
Figura 36 - Interface gráfica do GNS 3 e principais blocos	46
Figura 37 - Dispositivos disponíveis no GNS3.....	46
Figura 38 - Interface do VPCS	47
Figura 39 - Janelas de configuração de um <i>router</i>	48
Figura 40 - Arquitetura do sistema (baseada em [32, 61])	52
Figura 41 - Funcionamento do Sistema.....	56
Figura 42 - Funcionamento dos Agentes Móveis	57
Figura 43 - Funcionamento da Migração dos Agentes no Sistema	59
Figura 44 - Diagrama de tabelas.....	68
Figura 45 - Cenário criado no GNS 3.....	71
Figura 46 - Pings entre os computadores inseridos no cenário	73
Figura 47 - Comunicação entre máquinas virtuais no cenário	73

Figura 48 - Comunicação entre router "R1" e router do GECAD	74
Figura 49 - Cenário no GNS 3	74
Figura 50 - Interface Principal da Aplicação de Agentes.....	75
Figura 51 - Criação de um Agente na Aplicação de Agentes.....	76
Figura 52 - Resultados do AgentNA	76
Figura 53 - Trabalho do AgentPrinter bem sucedido.....	77
Figura 54 - Trabalho do AgentPrinter mal sucedido	77
Figura 55 - Resultados do AgentSwitch.....	78
Figura 56 - Visualização de relatórios antigos do AgentSwitch	78

Lista de Tabelas

Tabela 1 - Áreas Funcionais: Resumo.....	11
Tabela 2 - Classificação de agente	24

Notação e Glossário

API	<i>Application Programming Interface</i>
ASA	<i>Adaptive Security Appliance</i>
ASN.1	<i>Abstract Syntax Notation One</i>
ATM	<i>Asynchronous Transfer Mode</i>
DCE	<i>Data Circuit-terminating Equipment</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DNS	<i>Domain Name System</i>
DTE	<i>Data Terminal Equipment</i>
FCAPS	<i>Fault, Configuration, Accounting, Performance and Security</i>
GECAD	Grupo de Investigação em Engenharia do Conhecimento e Apoio à Decisão
GNS	<i>Graphical Network Simulator</i>
IAD	Inteligência Artificial Distribuída
IOS	<i>Internetwork Operating System</i>
IP	<i>Internet Protocol</i>
IPS	<i>Intrusion Prevention System</i>
ISEP	Instituto Superior de Engenharia do Porto
ISO	<i>International Organization for Standardization</i>
IT	<i>Information Technology</i>
MbD	<i>Management by Delegation</i>
MIB	<i>Management Information Base</i>
NAT	<i>Network Address Translation</i>
NVRAM	<i>Non-Volatile Random Access Memory</i>
OAA	<i>Open Agent Architecture</i>

In

OID	<i>Object Identifier</i>
OS	<i>Operating System</i>
OSI	<i>Open Systems Interconnection</i>
PC	<i>Personal Computer</i>
PIX	<i>Private Internet eXchange</i>
POS	<i>Packet Over SONET</i>
RAM	<i>Random Access Memory</i>
RFC	<i>Request for Comments</i>
RMON	<i>Remote Monitoring</i>
RPC	<i>Remote Procedure Call</i>
SMA	<i>Sistema Multiagente</i>
SMI	<i>Structure of Management Information</i>
SNMP	<i>Simple Network Management Protocol</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TMN	<i>Telecommunications Management Network</i>
UDP	<i>User Datagram Protocol</i>
VLAN	<i>Virtual Local Area Network</i>
VoIP	<i>Voice over Internet Protocol</i>
VPCS	<i>Virtual PC Simulator</i>
VPN	<i>Virtual Private Network</i>
XML	<i>EXtensible Markup Language</i>

1 Introdução

1.1 Enquadramento

A maioria das aplicações de gestão assentam no paradigma cliente-servidor, onde existe uma estação central de gestão que efetua a recolha e análise dos dados que são obtidos através de elementos de rede que se pretende gerir, e que se encontram fisicamente distribuídos. Neste paradigma são transferidas grandes quantidades de informação entre os elementos de rede que são alvos de gestão e a aplicação central, contribuindo para a diminuição da performance, uma vez que poderá conduzir a um elevado consumo de banda, bem como à possibilidade de ocorrer o chamado bottleneck na estação central de gestão. Este problema torna-se cada vez maior à medida que a rede cresce, assim como as exigências relativas aos serviços que esta suporta. O modelo tradicional, assente no paradigma cliente-servidor não é adequado para atender às exigências da realidade das redes atuais [1-4].

A solução para este problema passa pela utilização de abordagens descentralizadas de gestão, tais como *Remote Monitoring* (RMON), gestão por delegação (MbD), ou gestão baseada em agentes móveis [4-10].

Os agentes móveis podem ser usados para consultar e recolher dados das bases de dados com informação de gestão, de modo a fazer a monitorização do fluxo da rede num ambiente distribuído, assim como podem ser utilizados para realizar tarefas de configuração. Basicamente a solução assenta em distribuir o mecanismo de gestão com vista a ultrapassar as limitações de uma arquitetura centralizada.

Atualmente, na área da gestão de redes, tem sido feita muita investigação com o propósito de avaliar a aplicabilidade das tecnologias baseadas em agentes e agentes móveis na gestão das redes.

Introdução

Paralelamente, tem-se assistido a um aumento de investigação e estudo das várias plataformas que permitam o desenvolvimento de sistemas multiagente na gestão de redes [3, 4, 11-20].

Os agentes móveis podem ser utilizados para diminuir a sobrecarga nas comunicações da rede, através da utilização da delegação de autoridade, por parte da entidade responsável pela gestão, para um agente móvel. Os agentes são flexíveis, podem ser criados à medida dos requisitos do utilizador e lançados a partir de um gestor. Podem visitar cada elemento de uma rede, de acordo com uma tabela contendo o itinerário, processar os dados, analisá-los e se necessário efetuar funções de gestão, reportando posteriormente os resultados da sua atividade à aplicação gestora. Basicamente a solução assenta em distribuir o mecanismo de gestão de maneira a ultrapassar as limitações de uma arquitetura centralizada [3, 4, 21-23].

Esta abordagem permite, mover a inteligência para os nós onde os dados de gestão residem, uma vez que as decisões de gestão podem ser tomadas localmente, evitando a transferência de grandes quantidades de dados, desde os nós remotos até ao gestor central [4-24].

Seguindo esta abordagem foi desenvolvida uma aplicação de gestão baseada no paradigma dos agentes móveis para gerir a rede do grupo de investigação GECAD-ISEP. Foi proposta e desenvolvida uma aplicação multiagente para efetuar a gestão da rede, permitindo validar a aplicabilidade dos agentes móveis na área da gestão de redes.

No entanto, foram identificadas novas funcionalidades que deveriam ser adicionadas aos agentes propostos e desenvolvidos, assim como identificados novos agentes a serem incorporados na aplicação.

Paralelamente, o uso de simuladores permite representar virtualmente cenários reais onde é possível analisar e testar diversas abordagens e estudar diferentes configurações para uma rede de computadores. Neste trabalho é proposta a utilização de um simulador para representar a rede do GECAD, permitindo simular diversos cenários e efetuar análises aos resultados obtidos, com vista à validação da solução proposta.

1.2 Objetivos e Principais Contribuições

- Evoluir a aplicação de gestão *Mobile Agent Network Management Application*, nomeadamente, adicionando novas funcionalidades aos agentes móveis, já existentes, com vista à melhoria do seu desempenho;
- Propor e desenvolver novos agentes móveis a incorporar na aplicação de gestão *Mobile Agent Network Management Application* ;
- Investigar e propor um simulador de redes que permita a representação virtual da rede do GECAD e da aplicação de gestão *Mobile Agent Network Management Application*;

- Simular diversos cenários virtuais com vista a testar o trabalho proposto e desenvolvido.

1.3 Organização da Dissertação

Nesta secção é apresentada a estrutura deste documento, sendo este constituído por sete capítulos. De seguida é apresentada uma breve descrição de cada um dos capítulos:

- Neste primeiro capítulo, *Introdução*, é efetuado o enquadramento do tema deste trabalho, apresentados os principais objetivos e as contribuições;
- No segundo capítulo, *Introdução à Gestão de Redes*, pretende-se introduzir alguns conceitos fundamentais da gestão de redes, sendo apresentada a importância da gestão de redes, as suas áreas funcionais, o modelo arquitetural, o modelo de informação e o modelo relacional;
- No terceiro capítulo, *Gestão Baseada em Agentes*, apresenta-se uma primeira abordagem às definições de agente, tipos e classificação, sistemas multiagente, assim como o conceito de agentes móveis. Procura-se descrever as características e efetuar a comparação com a arquitetura cliente-servidor, entre outros;
- O quarto capítulo, *Simulador de Redes*, apresenta os simuladores de rede explicando a forma como funcionam. São introduzidos alguns conceitos chave sobre o tema e apresentadas algumas plataformas de simulação existentes no mercado. Segue-se a apresentação da plataforma escolhida para a realização deste trabalho, sendo expostas as razões da escolha efetuada, descritas as principais características e as vantagens que possui para a sua aplicação no trabalho aqui apresentado;
- No quinto capítulo, é feita uma descrição geral da aplicação *Mobile Agent Network Management Application*, apresentada a sua arquitetura, os agentes e respetivas funções, assim como a forma como os agentes interagem entre si. Para além disso, são apresentadas as suas principais limitações e de que forma estas podem ser ultrapassadas, bem como as novas funcionalidades que se pretende adicionar para evoluir a aplicação;
- No sexto capítulo, *Solução Proposta*, é apresentada a solução proposta e desenvolvida neste trabalho;
- As conclusões do trabalho desenvolvido nesta tese são apresentadas no sétimo e último capítulo, *Conclusões*. Neste capítulo são ainda descritas as principais limitações deste trabalho e apresentadas possíveis direções para trabalho futuro.

2 Introdução à Gestão de Redes

2.1 Introdução e Definições

A Gestão de redes é uma atividade essencial para garantir o funcionamento contínuo e eficiente de uma rede, e assegurar um elevado grau de qualidade dos serviços disponibilizados. Consiste numa ampla gama de funções, incluindo atividades, métodos, procedimentos, bem como o uso de ferramentas para administrar, operar e manter os sistemas e a rede em funcionamento de forma confiável. A operação consiste em manter a rede (e serviços a esta afetos) a funcionar sem problemas. Inclui a monitorização para detetar eventuais falhas de forma expedita, de preferência antes do utilizador final ser afetado. A administração envolve todos os recursos da rede e a forma como estão atribuídos, incluindo tarefas que permitam manter todos os recursos sob controlo. Quanto à manutenção, o nome já indica de forma implícita alguns dos conceitos que estão subjacentes, isto é, possibilita a realização de reparações, atualizações (sejam de *hardware* ou *software*) e ainda todas as operações necessárias para que a rede funcione corretamente quando sofre alterações significativas.

As três figuras seguintes, figura 1, 2 e 3, ilustram alguns dos conceitos base necessários para compreender o funcionamento da gestão de uma rede [2, 25, 26].

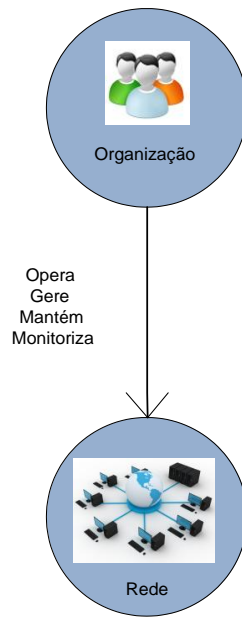


Figura 1- Uma Rede e a sua Organização

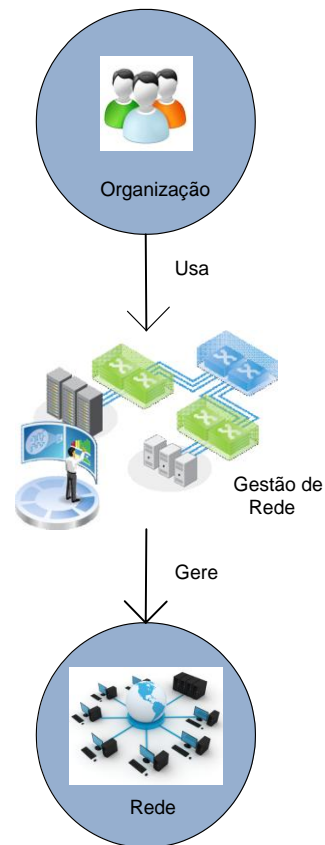


Figura 2- O papel da Gestão de Redes

Na figura 3 é possível observar todos os sistemas e aplicações que suportam as atividades e procedimentos que referimos anteriormente.

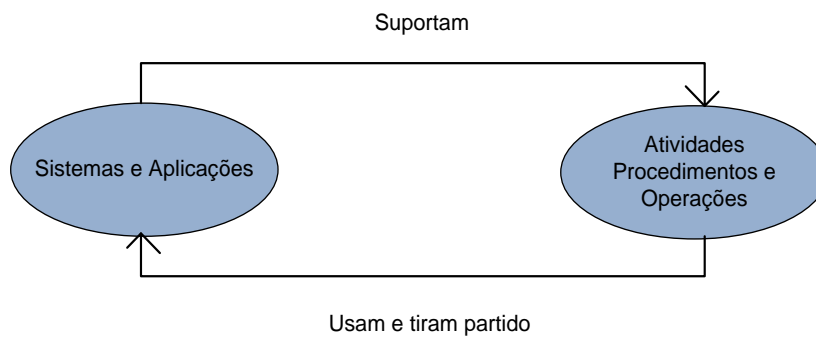


Figura 3- Componentes da Gestão de Redes

Na figura seguinte, figura 4, está representado o conceito de multidisciplinaridade, ou seja, para que seja possível fornecer um determinado serviço, normalmente existem vários componentes de rede envolvidos nessa função. Por norma, existe uma separação em termos de gestão de acordo com o ilustrado na figura 4.

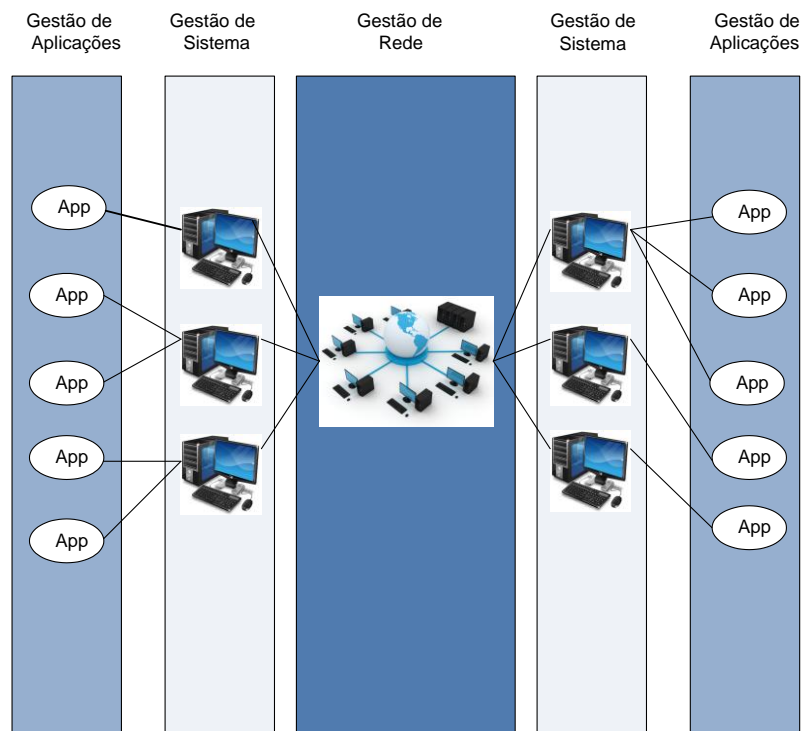


Figura 4- Gestão da Rede, Sistemas e Aplicações

2.2 Importância da Gestão de Redes

Nos dias de hoje a informação é vista pelas empresas como um aspeto fundamental para tirar vantagens competitivas sobre as organizações concorrentes. Esta circunstância fez com que o investimento nas tecnologias de informação tenha aumentado significativamente, aumentando a capacidade de tomar decisões de forma mais célere e segura [1].

Esta correlação entre os negócios e as redes faz com que a gestão destas seja crítica, uma vez que existe uma variante de recursos que são usados pelos utilizadores e que estes esperam que estejam disponíveis e funcionais de forma permanente, a qualquer dia e a qualquer hora. Para além destes serviços (DNS, DHCP, VPN, entre outros), existe um conjunto de aplicações, sistemas operativos, etc., que têm de funcionar de forma eficiente para não comprometer o negócio. Para tal, é necessária uma gestão rigorosa e eficiente.

As organizações vêm a sua produtividade ser aumentada quando as suas aplicações se encontram disponíveis e a funcionar, com um tempo de resposta ideal. Para ser credível, a quantificação destes benefícios tem de ser realista. Uma forma de fazer isto é olhar para as últimas dez a quinze interrupções que afetaram vários utilizadores e, caso a caso, fazer a pergunta: "Se eu tivesse o meu sistema de monitorização, esta interrupção poderia ter sido evitada ou resolvida de forma mais célere?".

Para qualquer uma destas interrupções ou falhas, uma forma de quantificar o impacto que existiu na produtividade da empresa pode ser calculado através da seguinte fórmula [2]:

- **Mão-de-obra perdida** (Duração da interrupção x Número de pessoas afetadas) x **Porcentagem da Produtividade Afetada = Impacto na Produtividade da Empresa**

A percentagem da produtividade afetada não vai ser 100% porque na maioria dos casos existem tarefas de substituição que são postas em curso para evitar que determinado recurso esteja parado.

Alternativamente, podemos usar a chamada regra "50-50" para calcular o impacto na produtividade da empresa. Esta regra teoriza que um sistema de monitorização bem planeado e administrado resultará no afastamento de 50% das interrupções do ano transato e reduzirá o tempo destas no ano corrente em 50% [2].

O que acontecerá analisando o impacto discutido nos parágrafos anteriores na produtividade das próprias pessoas ligadas aos departamentos de tecnologias de informação? Os resultados que normalmente surgem têm a ver com o facto de uma maior quantidade de trabalho ser realizada com o mesmo número de recursos humanos disponíveis, por oposição à necessidade de reduzir esse mesmo número. Naturalmente, surge a necessidade de justificar o investimento que é inevitável no aumento de trabalho realizado pelo grupo ligado à administração de sistemas informáticos. Esse tipo de cálculos pode ser realizado usando métodos parecidos com os apresentados no cálculo que o impacto das interrupções tinha na produtividade da empresa em geral. Basicamente, é necessário analisar junto da equipa de Tecnologias de Informação (IT) em cada elemento, qual a quantidade de trabalho feita de forma reativa vs planeada durante as interrupções, permitindo a inferência do nível de melhoramento de produtividade caso existisse um sistema de monitorização.

Falhas acontecem sendo indispensável detetar, diagnosticar e corrigir. O nível de serviços que foi previamente garantido aos utilizadores finais deve ser monitorizado e assegurado (por exemplo uma determinada largura de banda). Toda a panóplia de serviços, sua

implementação e disponibilização aos utilizadores finais deve ser gerida, incluindo quando estes são solicitados [2].

Depois das considerações feitas até aqui podemos ter a tentação de refutar toda esta importância subjacente à gestão de redes, uma vez que implementá-la não é fácil. É necessário saber o que existe e em que local, como tudo está interligado e conhecer fisicamente, e ao pormenor, cada um dos elementos de rede, e seus segmentos. No entanto, se um problema surge num dispositivo importante para o bom funcionamento da rede, torna-se indispensável o administrador ser avisado, e se possível serem efetuadas ações interventivas de forma a corrigir a falha antes de esta afetar os utilizadores. A simples ligação de um dispositivo com um IP que entre em conflito com outra máquina pode trazer problemas, assim como, a disponibilização involuntária de um serviço similar a um já existente na organização (ex. DHCP) [1, 2, 6].

Tempos de resposta e manutenção de serviços de forma eficiente resultam em vantagens competitivas. Um dos objetivos da gestão de redes é realizar as operações de forma eficiente. Se existirem ferramentas de teste e diagnóstico consegue-se identificar e resolver problemas de forma mais rápida. Paralelamente, existe a possibilidade de serem criados processos autónomos para determinados procedimentos, permitindo libertar os administradores de rede para outras funções que requeiram a sua atenção. É essencial analisar a performance da rede, assim como a existência de potenciais estrangulamentos. Este tipo de informação permite alocar recursos onde eles são realmente necessários, permitindo uma gestão adequada do investimento na infraestrutura. Outra das vantagens da utilização deste tipo de ferramentas é a facilidade existente na interação com a ferramenta. Após a mesma estar instalada e devidamente configurada qualquer utilizador é capaz de interpretar as informações e alertas dados pela mesma [2, 25].

Outro aspeto a ter em consideração nas redes atuais prende-se com a qualidade. Espera-se que exista um determinado nível de qualidade nas comunicações e nos serviços fornecidos. A garantia da qualidade implica monitorizar determinados parâmetros como por exemplo, a quantidade de banda disponível e o atraso nos tempos de resposta. A qualidade é fundamental para que o utilizador tenha confiança no desempenho da rede e nos serviços fornecidos.

2.3 Áreas Funcionais da Gestão de Redes

A *International Organization for Standardization* (ISO) sob a direção do grupo *Open Systems Interconnection* (OSI), criou um modelo de gestão de rede como o principal meio para a compreensão das principais funções dos sistemas de gestão de rede. Este modelo é conhecido como modelo OSI/ISO de gestão de rede e abrange cinco grandes áreas funcionais de sistemas de gestão de redes, às quais se dá vulgarmente o acrónimo de FCAPS (*Fault, Configuration, Accounting, Performance e Security*). É apresentado de seguida um diagrama ilustrativo das áreas funcionais [3, 22, 27-29]:

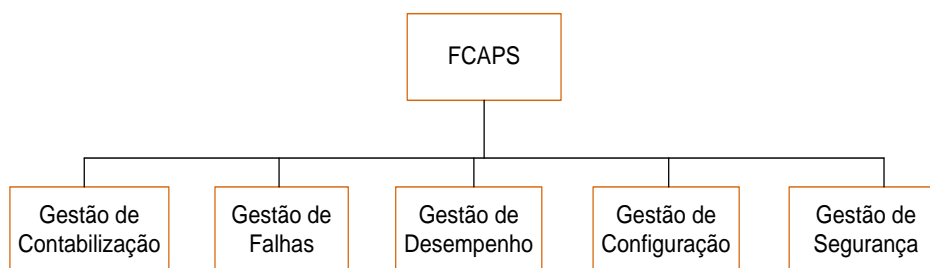


Figura 5- As cinco áreas funcionais da gestão de redes do modelo OSI

A **gestão de falhas** é a área mais amplamente implementada devido ao facto de as falhas poderem causar períodos de inatividade e influenciar negativamente a operação da rede. Tem como objetivo reconhecer, isolar, corrigir e registar falhas que ocorrem na rede. Deteta, notifica e corrige automaticamente alguns problemas de rede de forma a manter a mesma em funcionamento e de forma eficiente. O processo de gestão de falhas envolve:

- Determinar os sintomas e isolar o problema;
- Resolver o problema e testar a solução;
- Gravar a deteção e resolução do problema.

Através da gestão de falhas, o administrador de rede pode de uma forma pró-ativa e/ou reativa identificar rapidamente uma falha e iniciar imediatamente o processo de solução/resolução.

A **gestão de configuração** caracteriza-se pelo conjunto de operações necessárias para a inicialização, término, alteração e armazenamento da configuração dos equipamentos de rede. Permite obter controlo, supervisão e coleta de informações sobre mudanças de configuração em cada um dos objetos geridos. Esta área é especialmente importante, uma vez que muitos problemas de rede surgem como consequência de alterações efetuadas em ficheiros de configuração, versões de *software*, ou alterações de *hardware* do sistema. Através da gestão de configuração, o administrador consegue alterar facilmente a configuração dos equipamentos, aceder a documentação sobre a configuração dos mesmos, manter um inventário atualizado, bem como monitorizar os recursos existentes e os seus detalhes.

A **gestão de contabilização** permite identificar e atribuir custos relativamente à utilização dos recursos, como por exemplo, espaço em disco, tempo de conexão, quantidade de conexões, tempo de processamento, utilização de largura de banda, entre outros. Desta forma podemos saber quantos e quais os recursos a ser consumidos, por quem e a que custo. A gestão de contabilização é igualmente útil para estabelecer limites de utilização, para se necessário atribuir cotas a grupos de utilizadores, utilizadores individuais e estabelecer planos de tarifas aos recursos.

A **gestão de desempenho** está relacionada com a coleta de dados para analisar e avaliar o desempenho da rede relativamente a tempos de resposta, taxas de perda de pacotes,

utilização, entre outros. Ou seja, é representada pelo conjunto de funções necessárias para o administrador monitorizar e analisar as atividades na rede, efetuando (se necessário) alterações e ajustes na mesma, para que esta se mantenha a funcionar a níveis aceitáveis. Monitorizar ativamente o desempenho da rede é um passo importante na identificação de problemas antes que eles ocorram, como parte de uma estratégia proactiva de gestão da rede.

A **gestão de segurança** tem como objetivo controlar o acesso aos ativos de rede. Não se preocupa apenas em garantir a segurança do ambiente de rede, mas também em analisar regularmente informação relacionada com a segurança. Entre as várias funções e tarefas que estão relacionadas com esta área de gestão, podemos dar como exemplos a autenticação na rede, autorizações de acesso, configuração e gestão de *firewalls*, sistemas de deteção de intrusos, políticas de segurança e listas de acesso. Trata-se do conjunto de funções que o administrador de rede deve executar para identificar e proteger equipamentos e dados da rede de ataques e violações oriundas de pessoas não autorizadas.

A tabela seguinte, tabela 1, apresenta um pequeno resumo das principais funções de cada área funcional, com vista a se conseguir uma gestão correta e eficiente da rede [3, 22, 27-29]:

Tabela 1 - Áreas Funcionais: Resumo

Gestão de Falhas	Gestão de Configuração	Gestão de Contabilização	Gestão de Desempenho	Gestão de Segurança
Deteção de Falhas	Inicialização de Recursos	Monitorização de Serviços	Garantir Nível de performance	Acesso a Recursos de forma Seletiva
Correção de Falhas	Configuração Remota	Monitorização do Uso de Recursos	Recolha de Dados	Logs de Acesso
Isolamento de Falhas	Inicialização de Tarefas e sua monitorização	Custo por Serviço	Geração de Relatórios	Privacidade dos Dados
Recuperação da Rede	Suporte para Instalação Automática de Software	Combinação de Custos para Vários Recursos	Análise de Dados de Performance	Verificação de Permissões de Acesso
Geração, Análise e Tratamento de Alarmes	Gestão de Inventário	Utilização de Quotas	Planeamento/Ajuste do Sistema	Geração de Alarmes
Testes de Diagnóstico	Mudança de Configurações	Verificação de Adulteração de Tarifas	Manutenção de Logs e Dados Históricos	Tratamento de Falhas de Segurança
Análise, Tratamento e Logging de Erros	Término de Recursos	Auditorias	Análise de Taxas de Erros	—

2.4 Modelo Arquitetural da Gestão de Redes

O modelo arquitetural da gestão de redes é utilizado para a descrição das entidades que estão envolvidas em tarefas de gestão, bem como as suas interfaces e métodos de comunicação.

Dos vários elementos existentes na gestão de redes é de salientar a existência de dois que são fundamentais: o gestor de redes e o objeto gerido. O primeiro consiste no *software* e *hardware* responsáveis pela coleta, análise e processamento da informação de gestão. Nos dias de hoje existem várias ferramentas disponíveis no mercado que podem desempenhar este papel, algumas pagas e outras *freeware*. Quanto ao segundo, consiste numa abstração (uma vez que pode estar relacionado com comunicações, processamento de dados, recurso físico, etc.) de um determinado recurso alvo de gestão. De salientar que quando nos referimos a recurso, não devemos entender uma determinada máquina. Para gerir um determinado recurso é necessário aceder à sua base de dados com informação de gestão, MIB, através de um agente de gestão, que comunica e processa a informação de gestão. Se existir comunicação entre o gestor de redes e o objeto gerido é necessário que exista também um protocolo de gestão, pois é quem define as regras de comunicação entre ambos os elementos, assim como garante que seja possível obter e interpretar cada mensagem enviada/recebida [1, 6, 22].

Resumidamente, a arquitetura de gestão tem de conseguir espelhar a forma como os objetos geridos e o gestor interagem e comunicam, entre outros, para fins de gestão.

2.4.1 Arquiteturas Centralizadas

Este tipo de arquitetura é o mais utilizado nos vários modelos de gestão, nos quais se destacam o OSI, TMN, SNMPv1, SNMPv2 e SNMPv3. Esta arquitetura assenta na existência de uma topologia Gestor-Agente, onde existe apenas uma entidade responsável pela gestão em conjunto com os restantes objetos geridos [3, 6, 22]. Na figura seguinte, figura 6, é possível verificar um exemplo deste tipo de arquitetura.

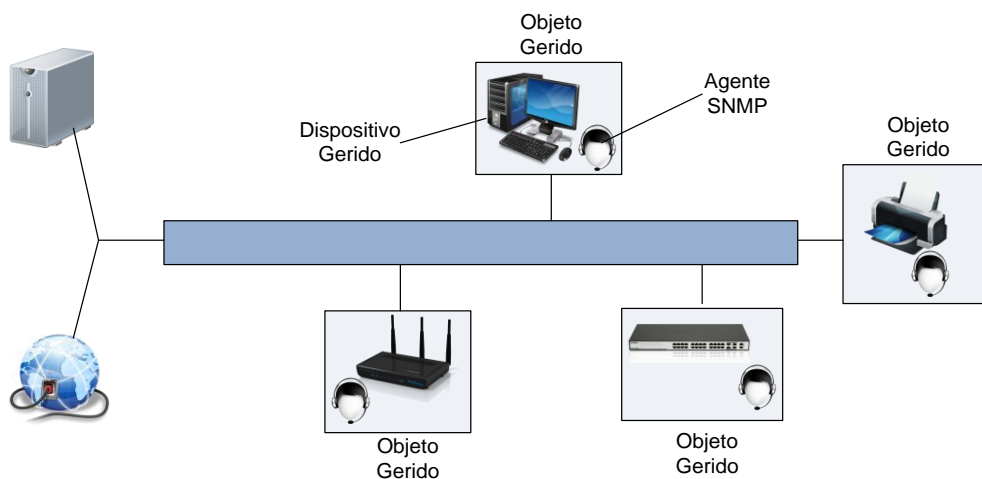


Figura 6- Exemplo de Arquitetura Centralizada (baseado em [6])

Cada objeto gerido precisa de interagir com o gestor de rede, sendo esta interação efetuada através de agentes que detêm a capacidade de realizar consultas à MIB do objeto gerido e compreender a informação que lá se encontra. Estes agentes estão associados a cada objeto gerido tendo como função permitir a gestão dos dispositivos e/ou serviços que estes disponibilizam. Não possuem qualquer tipo de "inteligência", limitando-se apenas a permitir a recolha remota dos dados de gestão e a executar as ações que são propostas pela estação de gestão. Assim, a estação de gestão é única e concentra em si todas as funcionalidades e a capacidade de processar a informação de gestão [3, 6, 22].

As arquiteturas centralizadas apresentam como grande vantagem o facto de serem relativamente simples de desenhar e implementar, sendo os dispositivos representados pelas suas MIBs e o processamento centralizado. No entanto, possuem desvantagens, tais como a flexibilidade reduzida e a reconfiguração limitada.

Este tipo de arquitetura obriga a que uma grande quantidade de tráfego de gestão atravesse a rede, podendo em alguns casos causar um *bottleneck* quer no segmento em que se encontra o gestor de redes, quer nos elementos de rede com que este interage.

2.4.2 Arquiteturas Descentralizadas

O facto de as redes atuais possuírem grandes níveis de complexidade e grandeza, faz com que as arquiteturas centralizadas sejam consideradas desadequadas uma vez que, como já foi referido anteriormente, estas arquiteturas apresentam problemas de escalabilidade e flexibilidade. Este aumento de complexidade (e conseqüente necessidade de escalabilidade), em conjunto com as desvantagens da gestão centralizada, contribuem para que a gestão distribuída ganhe um maior relevo, sendo possível melhorar a escalabilidade, a tolerância a falhas, bem como permitir realizar processamento local de dados, gerando assim mais valor acrescentado [3, 6, 7, 22, 30].

Nesta arquitetura parte das tarefas de gestão são distribuídas pelos agentes, atribuindo-lhes desta forma uma certa "inteligência". O grau e tipo de descentralização dependem das funcionalidades que o administrador desejar. A arquitetura distribuída permite assim manipular grandes quantidades de informação de gestão de forma mais eficiente, sendo que os dados gerados dessa manipulação e transferidos para a rede para efeitos de comunicação serão menores [3, 6, 7, 22, 30]. Beneficia sistemas exigentes, de grande complexidade e que exigem grande interação, uma vez que estão mais próximos da informação e resolvem o problema do *polling* (consulta aos agentes), obrigando a que os cálculos sejam efetuados localmente.

A forma mais simplificada de descentralização é a monitorização remota RMON (*Remote Network Monitoring*). São utilizados dispositivos para analisar o tráfego da rede como um todo e atuam como "sondas remotas" (tipo *sniffers*) que analisam os pacotes de rede baseando-se num conjunto de premissas (configuradas remotamente), filtram e processam informação. Se for detetado algum problema, têm a capacidade de tomar ações interventivas

como por exemplo enviar alertas para o gestor de redes. O RMON foi o primeiro agente SNMP "parametrizável" via SNMP e com capacidade de processamento significativa, ou seja, já não são "agentes estúpidos" [3, 6, 7, 22, 30].

As arquiteturas distribuídas (ou descentralizadas) podem ser divididas em três tipos: Arquitetura Hierárquica, Cooperativa e Altamente Distribuída.

No primeiro tipo, a delegação de tarefas é feita verticalmente e estas são executadas do nível superior para o inferior. É caracterizada por possuir uma estrutura de gestores em árvore, figura 7, ficando os agentes no nível mais baixo.

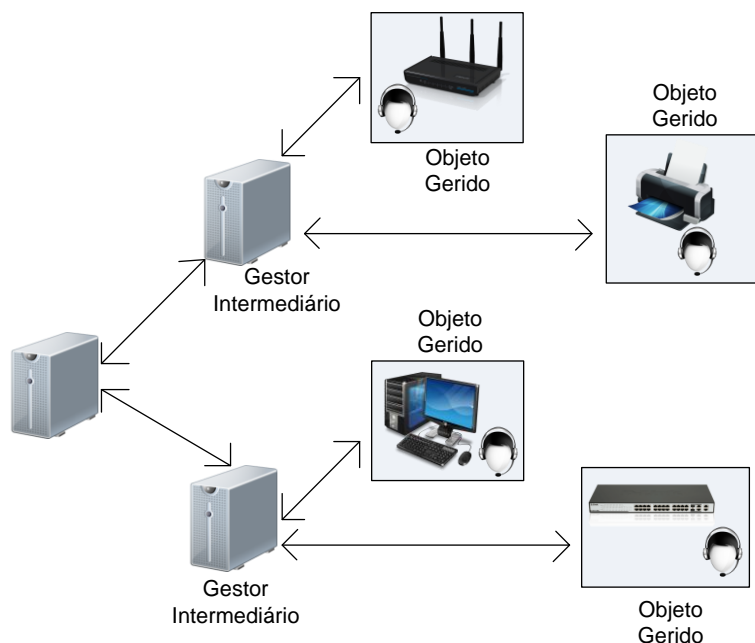


Figura 7- Arquitetura Hierárquica (baseada em [6])

A aplicação de gestão delega as tarefas para os níveis inferiores, podendo ser agentes ou gestores intermédios. Estes gestores recebem as diligências do nível superior e executam-nas sobre os níveis inferiores, sendo o controlo efetuado superiormente. Os agentes apenas contêm os objetos geridos sobre os quais são executadas as tarefas de gestão [6, 22].

A arquitetura Cooperativa assenta no conceito da gestão por objetivos. Os agentes são autónomos e interagem entre si com o intuito de atingir os objetivos que foram anteriormente definidos. Na figura seguinte, figura 8, é possível verificar um exemplo desta arquitetura, onde é lançado um programa na rede (agente móvel) por parte do gestor, para percorrer um determinado caminho (previamente definido). À medida que esse caminho vai sendo percorrido, o agente móvel recolhe e processa a informação de gestão dos vários elementos. Quando regressa ao ponto de partida, entrega um relatório ao gestor que contém os resultados das suas ações durante o percurso efetuado. Esta é a abordagem que foi utilizada para desenvolver a aplicação *Mobile Agent Network Management Application* que serve de base para o desenvolvimento deste trabalho.

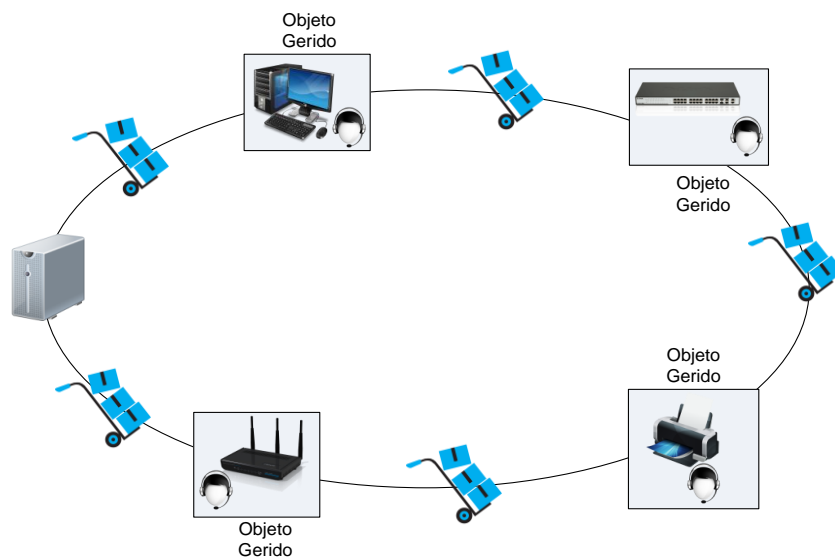


Figura 8- Arquitetura baseada em Agentes Móveis (baseada em [6])

Esta arquitetura é mais complexa, mas também mais flexível. No entanto, obriga a que os dispositivos possuam uma capacidade de processamento razoável para que possa ser implementada, ou seja, é exigente relativamente a recursos.

Trata-se de uma arquitetura altamente distribuída, que tem como base a arquitetura hierárquica, quer na delegação de tarefas em termos verticais como funcionais. Verifica-se por outro lado uma arquitetura cooperativa entre pares de gestão, permitindo assim interação em termos de gestão. A figura seguinte, figura 9, é um exemplo de uma arquitetura altamente distribuída.

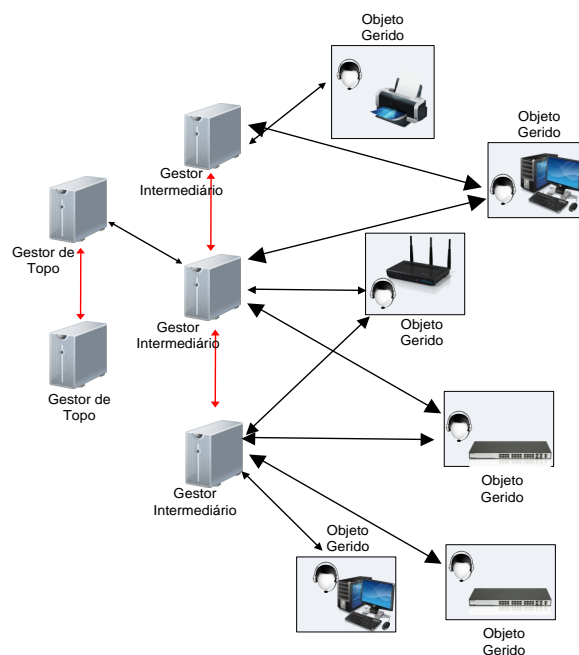


Figura 9- Arquitetura altamente distribuída [38]

Devido à estrutura que implementa, esta arquitetura fornece alta disponibilidade no que à gestão diz respeito. É uma abordagem bastante escalável o que permite maior desagregação de funções, possibilitando uma diminuição da tendência de sobrecarga dos sistemas. No entanto, a arquitetura altamente distribuída apresenta algumas desvantagens, nomeadamente:

- Aumento de tráfego de gestão nos dispositivos geridos, uma vez que dão resposta a várias estações de gestão ao mesmo tempo;
- Os agentes necessitam de fornecer a mesma informação a mais que uma estação de gestão, aumentando as tarefas de gestão a desempenhar. Isto tem reflexo no tempo de processamento de gestão e tráfego de rede.

2.5 Modelo de Informação de Gestão

Este modelo é o responsável pela representação lógica dos recursos que são geridos, isto é, tem a responsabilidade de definir de forma pormenorizada como é feito o acesso à informação de gestão, que significado tem essa informação de gestão e como descrever nova informação de gestão [1, 2, 6, 22, 26, 28, 31, 32].

A MIB pode ser vista como um repositório que contém a informação de gestão existente no objeto gerido. Ou seja, funciona como um contentor que cada objeto gerido possui, onde se encontra toda a informação de gestão relativa a esse objeto. Portanto, as operações de gestão são efetuadas neste contentor, sendo os objetos geridos representados em forma de tabela, por exemplo. De salientar que a MIB não é uma base de dados real, mas sim uma forma de vermos um dispositivo.

A construção das estruturas da MIB é feita de acordo com regras que são descritas pela *Structure of Management Information - SMI (RFC 1155 e RFC 2578)*, que consiste num conjunto de documentos que definem: a forma de identificação e grupos de informação; sintaxes permitidas; tipos de dados permitidos. Esta estrutura além de definir a organização com a qual a MIB deve ser criada, torna simples a definição de novas MIBs [33]. Inclui vários parâmetros, mais concretamente:

- Tipos de dados e a sua representação;
- Relação entre cada objeto gerido e a sua representação;
- Relação entre o objeto gerido e a sua descrição de acordo com a *SMI Internet*. Para tal, emprega um subconjunto da sintaxe de *ASN.1 (Abstract Syntax Notation One)* [32].

Na SMI os objetos estão organizadas em árvore e são identificados (e conhecidos) por um *Object Identifier* (OID). Este identificador é constituído por um conjunto de números que é obtido a partir dos nós das árvores. Na figura seguinte, figura 10, é possível ver a árvore da MIB II, sendo que o OID *internet* corresponde a *iso.org.dod.internet* ou *1.3.6.1* [32].

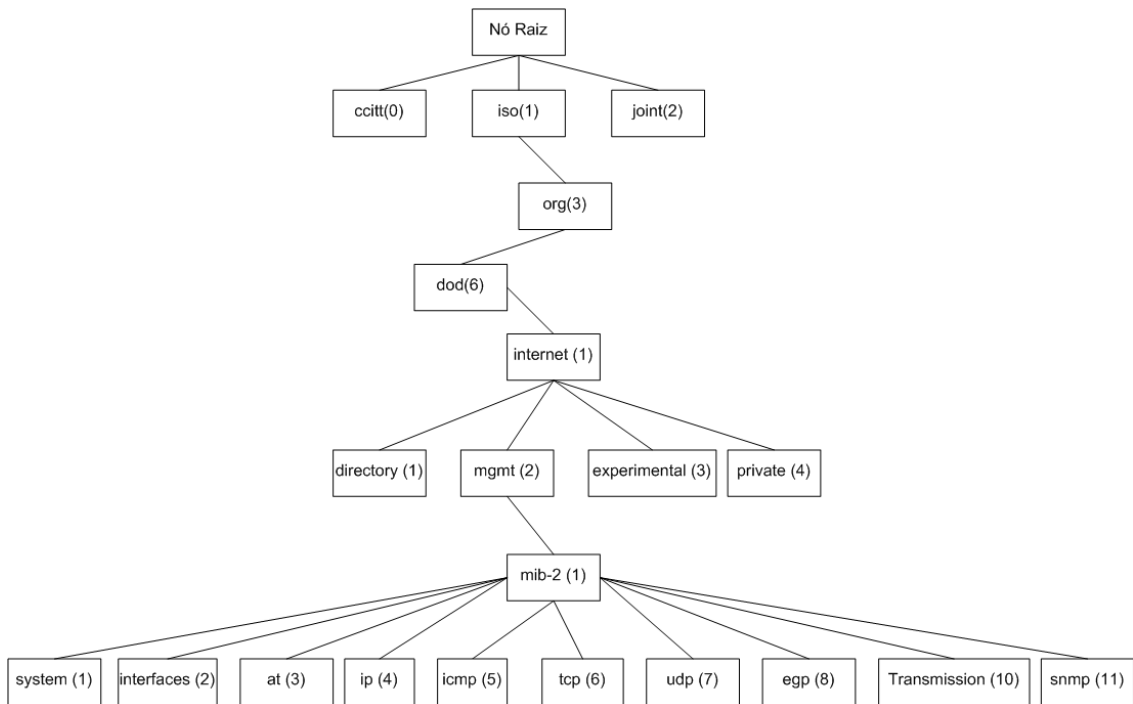


Figura 10- MIB II [32]

Hoje em dia existem várias ferramentas disponíveis que permitem efetuar consultas a uma MIB de uma forma simples e intuitiva. Um exemplo de uma dessas ferramentas é a "*iReasoning Mib Browser*" disponibilizada pela *iReasoning SNMP API*. Na próxima figura, figura 11, é apresentada a interface gráfica da ferramenta onde é visível alguma informação de gestão sobre um determinado elemento de rede gerido [34,35].

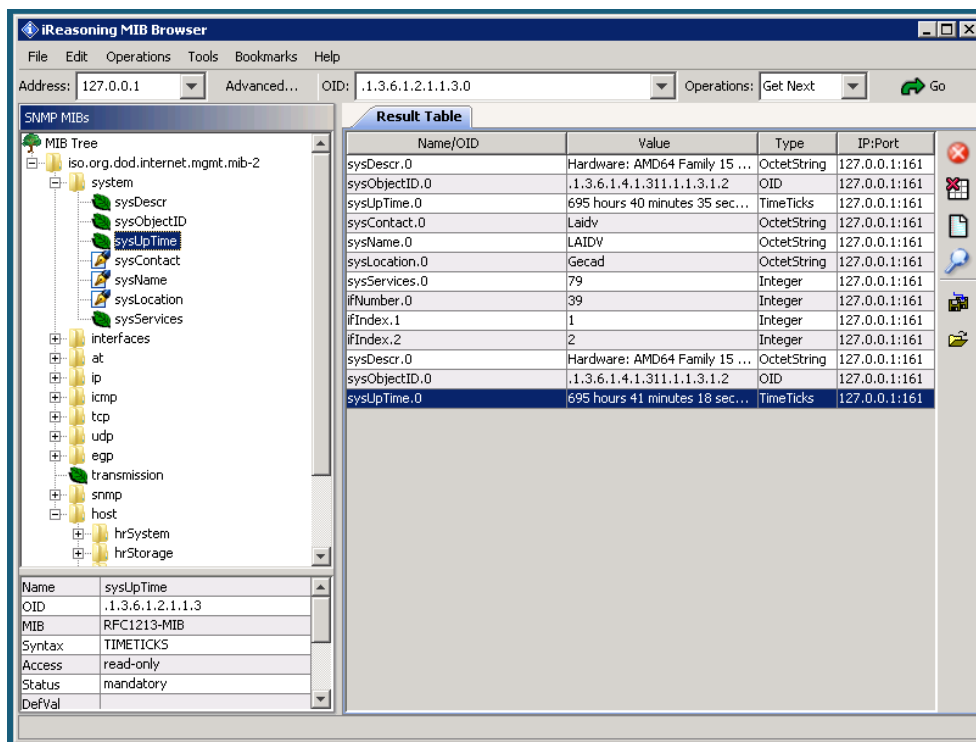


Figura 11- MIB Browser

Ao analisarmos a figura é possível verificar o endereço IP do dispositivo em causa, neste caso o endereço de *localhost* 127.0.0.1. Ao seu lado direito encontra-se o OID da MIB que é o alvo de gestão. Do lado esquerdo é representada a árvore da MIB, onde é possível explorar todos os OIDs que existem na mesma, e ao centro encontra-se a tabela com os dados resultantes das operações de gestão, nomeadamente as consultas.

De seguida é apresentada uma figura, figura 12, onde é possível verificar os resultados de uma consulta efetuada através de uma operação de gestão chamada "GET-BULK".

Name/OID	Value	Type	IP:Port
sysDescr.0	Hardware: AMD64 Family 15 Model 65 Stepping 3 AT...	OctetString	127.0.0.1:161
sysObjectID.0	.1.3.6.1.4.1.311.1.1.3.1.2	OID	127.0.0.1:161
sysUpTime.0	696 hours 2 minutes 33 seconds (250575315)	TimeTicks	127.0.0.1:161
sysContact.0	Laidv	OctetString	127.0.0.1:161
sysName.0	LAIDV	OctetString	127.0.0.1:161
sysLocation.0	Gecad	OctetString	127.0.0.1:161
sysServices.0	79	Integer	127.0.0.1:161
ifNumber.0	39	Integer	127.0.0.1:161
ifIndex.1	1	Integer	127.0.0.1:161
ifIndex.2	2	Integer	127.0.0.1:161
ifIndex.3	3	Integer	127.0.0.1:161
sysUpTime.0	696 hours 7 minutes 38 seconds (250605834)	TimeTicks	127.0.0.1:161
sysContact.0	Laidv	OctetString	127.0.0.1:161
sysName.0	LAIDV	OctetString	127.0.0.1:161
sysLocation.0	Gecad	OctetString	127.0.0.1:161
sysServices.0	79	Integer	127.0.0.1:161
ifNumber.0	39	Integer	127.0.0.1:161
ifIndex.1	1	Integer	127.0.0.1:161
ifIndex.2	2	Integer	127.0.0.1:161
ifIndex.3	3	Integer	127.0.0.1:161
ifIndex.4	4	Integer	127.0.0.1:161

Figura 12- Tabela resultante da operação "GET-BULK"

Analisando a tabela é possível ver que esta é constituída por quatro colunas, mais concretamente:

- A coluna *Name/OID* corresponde à OID que estamos a consultar, que contém também uma semântica própria;
- A segunda coluna, *Value*, representa o valor contido na OID;
- A coluna *Type* consiste no tipo de dados do valor da OID;
- Por último, *IP-Port*, que como o nome indica contém o IP do elemento e a porta por onde comunica o SNMP.

2.6 Modelo Relacional de Gestão

O modelo relacional de gestão tem com responsabilidade descrever como é que os sistemas gestor e entidades geridas comunicam entre si. De seguida é apresentado o protocolo SNMP, bem como outros protocolos utilizados na gestão.

2.6.1 SNMP

O protocolo SNMP é um exemplo de protocolo de gestão de redes IP baseado no modelo *Internet*.

Foi desenvolvido para redes IP nos anos 80 e hoje em dia encontra-se na terceira versão (SNMPv3). A sua função consiste em efetuar/permitir toda a comunicação de informação de gestão entre o gestor e a entidade gerida, e em alguns casos entre gestores. Ou seja, de uma forma muito simples pode-se afirmar que este protocolo permite aos administradores de rede gerir os equipamentos da rede e diagnosticar os seus problemas.

A arquitetura do SNMP é constituída por quatro elementos principais:

- Gestor;
- Agentes;
- MIBs;
- SNMP (para a comunicação).

O **gestor** é a entidade responsável por implementar todas as tarefas necessárias para efetuar a gestão da rede propriamente dita. É nesta entidade que reside toda a capacidade de processamento e analisada a informação de gestão recolhida nos agentes. É também a

entidade responsável por tomar decisões de gestão. Deve possuir uma interface simples e intuitiva.

Quanto aos **agentes**, estes são os elementos que representam as entidades geridas. Devem ser simples e não necessitar de uma grande quantidade de recursos de *software* e *hardware* para funcionar corretamente. Estes são implementados normalmente em dispositivos de rede como *routers*, *switches*, placas de rede, etc., e respondem (enviam informação de gestão para o gestor) a pedidos efetuados pelo gestor.

Relativamente às **MIBs**, e como já foi referido anteriormente, possuem um conjunto de objetos que caracterizam os objetos geridos.

Finalmente, o próprio protocolo **SNMP** garante que seja possível a comunicação entre o gestor e os agentes. Portanto, este protocolo define as regras (a nível semântico e sintático) de todas as mensagens passíveis de serem trocadas entre as entidades. No entanto, este conjunto de mensagens é bastante limitado devido à simplicidade do protocolo em causa.

Na figura seguinte, figura 13, é apresentada a arquitetura do protocolo SNMP onde está ilustrada a forma como os diversos elementos da arquitetura estão distribuídos, assim como a é feita a comunicação entre eles.

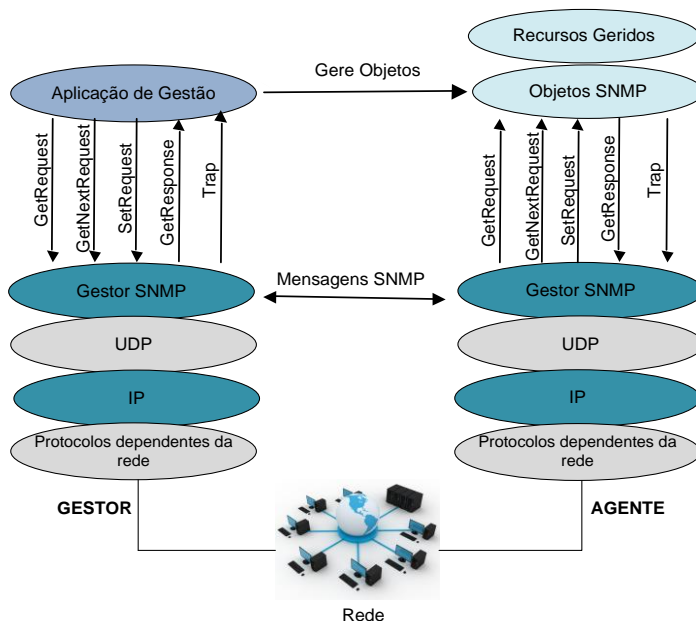


Figura 13 - Arquitetura do SNMP ([baseada em 36])

Como podemos verificar na figura anterior a aplicação de gestão bem como os agentes utilizam os comandos *GetRequest*, *GetNextRequest*, *SetRequest*, *GetResponse* e *Trap* para realizarem as suas transações de informação. Nas versões mais recentes do protocolo, SNMPv2 e SNMPv3, existe ainda uma operação protocolar conhecida como “*GetBulkRequest*”, sendo que todas estas mensagens são respondidas com “*GetResponse*” por parte do agente. Uma chamada de atenção para a operação *Trap*, que consiste numa operação protocolar

assíncrona gerada pelo agente em situações excepcionais pelo porto UDP 162. Este envio surge quando o agente deteta um determinado problema num objeto gerido (por exemplo uma partição do disco com espaço reduzido), contactando assim o gestor [1, 3, 6, 22, 28, 36].

No protocolo SNMP a segurança sempre foi alvo de grande discussão, pois, a primeira versão continha permissões baseadas apenas numa palavra passe (*community string*), que era enviada em cada mensagem, em texto claro. Só se verificaram alterações relativamente às questões da segurança na última versão do protocolo (SNMPv3), onde as mensagens passaram a ser encriptadas. Esta modificação na segurança do protocolo permite alcançar defesas contra alguns ataques de segurança mais comuns, nomeadamente: alteração da informação, espionagem, alteração da ordem das mensagens, etc. No entanto, ainda não foram criados mecanismos de proteção para alguns tipos de ataques, nomeadamente para a negação de serviço, o que pode colocar em causa a comunicação entre agente e gestor.

A figura seguinte, figura 14, ilustra os princípios fundamentais da arquitetura, na versão SNMPv3, abordando-a numa perspetiva de entidade e módulos. Estes dois conceitos devem ser diferenciados e explicados: a arquitetura de gestão SNMP é constituída por um conjunto de entidades SNMP distribuídas, que cooperam entre si, podendo funcionar como gestor, agente ou ambos; cada uma das entidades é estabelecida tendo por base módulos, que interagem entre si para providenciar serviços [1, 3, 6, 22, 28, 36, 37].

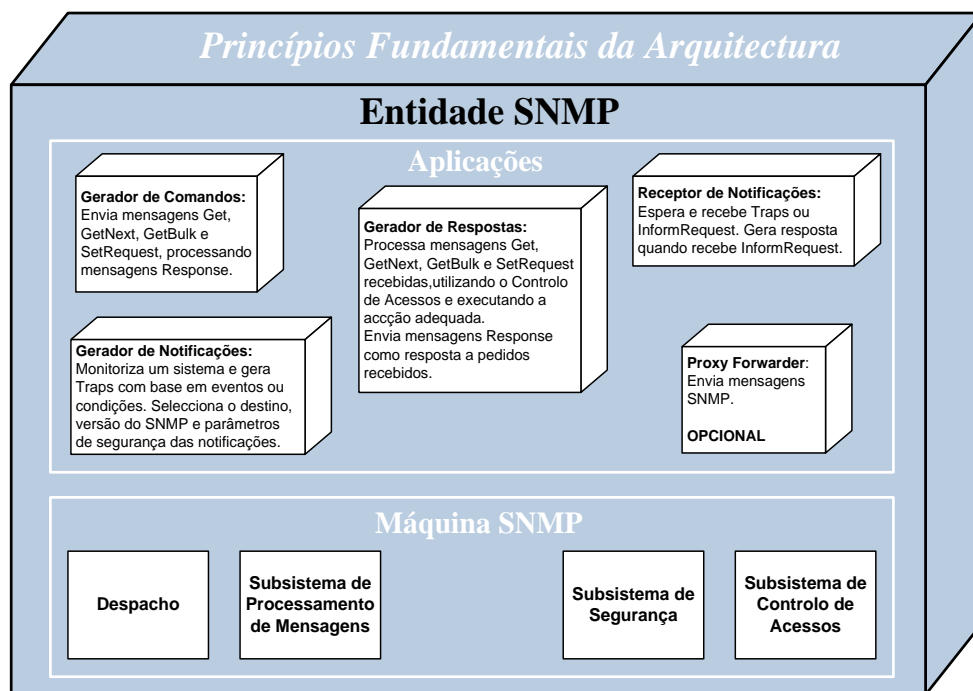


Figura 14- Entidade SNMP [32]

2.7 Conclusões

A monitorização da rede bem como a deteção e resolução de eventuais problemas de gestão que possam surgir podem ter várias abordagens que vão desde como lidar com um problema de forma individualizada até uma visão integrada da infraestrutura, atuando sobre ela como um todo.

Ao atuar sobre toda a infraestrutura, as arquiteturas de gestão assumem um papel importantíssimo uma vez que constituem a base para o desenvolvimento de sistemas de gestão abertos, e aplicáveis nos ambientes heterogéneos existentes nas atuais infraestruturas informáticas.

3 Gestão Baseada em Agentes

3.1 Definição de Agente

Atualmente existem várias definições no que se refere a agentes. Pode-se dizer que um agente para além de ser um paradigma de comunicação é uma entidade física, ou virtual, que funciona de forma contínua e autónoma num determinado ambiente, que poderá também ser habitado por outros agentes e processos [38]. Os agentes têm a capacidade de entender esse ambiente, comunicar com os outros agentes e ganhar autonomia, como resultado das interações com os outros agentes da comunidade. O agente pode ser visto como uma entidade cognitiva, ativa e autónoma que consegue operar com o meio envolvente e com outros agentes sem algo ou alguém para o aconselhar e controlar. Portanto, os agentes além de capacidade para raciocinar, também têm a capacidade de agir, ações estas que afetam o meio ambiente, que, por sua vez, afeta as decisões futuras dos agentes [38-42].

Como foi referido no início desta secção, existem várias definições relativamente aos agentes. Um focam-se no agente como entidade individual, como por exemplo um agente que realiza uma determinada ação (como uma reserva) em nome de uma determinada entidade, outras, são mais orientadas para as competências dos agentes sociais. Enquanto que no primeiro caso a abordagem está associada ao conceito de agente inteligente, no segundo caso, existe uma associação ao conceito de Sistemas Multiagente (*Multiagent System*), uma vez que o foco não é o agente em si, mas a comunidade dos agentes e portanto, as características mais valorizadas são a argumentação, negociação e capacidade de resolução de conflitos [38-42].

3.2 Tipos e Classificação de Agentes

Os agentes podem ser classificados de acordo com as suas características. Na tabela que é apresentada de seguida, tabela 2, é possível verificar um conjunto de características que podem ser utilizadas para descrever/classificar um agente [39-42].

No capítulo 5 serão apresentadas e discutidas as características dos agentes que consideramos serem mais importantes no contexto deste trabalho, mais concretamente, na gestão de redes baseada em agentes móveis.

Tabela 2 - Classificação de agente

Propriedades	Significado
Capacidade Sensorial	Possui “sensores” de forma a reunir informação sobre o ambiente
Reatividade	Sente e age, reagindo às constantes mudanças do ambiente
Autonomia	Decide e controla as suas próprias ações
Pro-Atividade	É orientado a objetivos, indo além do simples reagir ao ambiente
Persistência	Existe durante longos períodos de tempo
Capacidades Sociais	Comunica e coopera com outros agentes ou mesmo pessoas, concorre, negocia, etc.
Aprendizagem	Capaz de mudar o comportamento baseando-se em experiências anteriores
Mobilidade	Capaz de mover-se de um computador para outro
Flexibilidade	As suas tarefas não precisam de estar obrigatoriamente pré-determinadas
Agilidade	Capacidade de rapidamente tirar proveito de novas oportunidades
Carácter	Comportamento emocional e personalidade credível
Inteligência	Capacidade de raciocinar de forma autónoma, para planear as suas ações, corrigir os seus erros, para reagir a situações inesperadas, adaptar e aprender

3.3 Sistemas Multiagente

Os Sistemas Multiagente (SMA) consistem em sistemas constituídos por vários agentes inseridos num ambiente que lhes é comum, onde os agentes detêm um comportamento autónomo, mas que ao mesmo tempo interagem com os outros agentes que se encontram no mesmo ambiente. Estes agentes possuem duas características essenciais, mais concretamente: a capacidade de agir autónoma, tomando decisões para satisfazer os seus objetivos; serem capazes de interagir com outros agentes através de protocolos de interação social inspirados nos humanos, e onde são incluídas funcionalidades como a coordenação, competição e negociação [42]. Portanto, um SMA é um sistema computacional em que dois ou mais agentes homogéneos ou heterogéneos trabalham em equipa para realizar determinadas tarefas e atingir objetivos específicos. Para que cada um destes agentes possa operar como parte do

sistema, é necessária uma infraestrutura que permita a comunicação e/ou cooperação entre os agentes que compõe o SMA. Na figura seguinte, figura 15, é apresentada essa mesma infraestrutura.

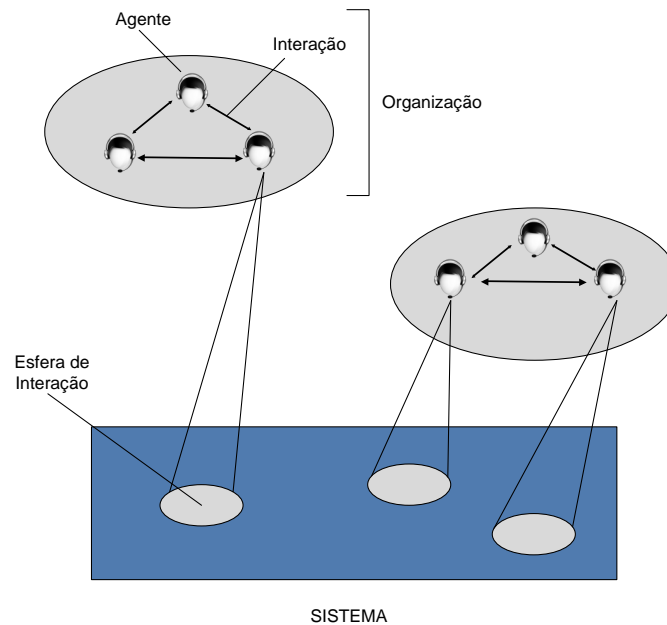


Figura 15 - Estrutura de um SMA (baseada em [42])

Ao analisar a figura anterior é possível verificar que o SMA é constituído por vários agentes, onde cada um tem a capacidade de percepção e ação no sistema. Cada agente tem também uma esfera de influência distinta sobre o sistema (ambiente), isto é, é capaz de influenciar diferentes partes do sistema [42], podendo existir coincidências entre estas partes, dependendo das relações existentes entre os agentes. Por exemplo, num SMA que tenha como objetivo gerar horários de vários departamentos numa universidade, a esfera de influência vai recair sobre os horários dos docentes, turmas e salas desses departamentos. Com relativa facilidade pode dar-se o caso de dois agentes interagirem entre si, no caso de efetuarem alocações em docentes, turmas ou salas que lhes sejam comuns [39-42].

A área dos Sistemas Multiagente derivou de uma outra área, a Inteligência Artificial Distribuída (IAD). A IAD não tem como foco principal a coordenação de processos concorrentes ao nível da resolução de problemas ou representação, nem sistemas operativos distribuídos. A IAD foca-se essencialmente na resolução de problemas onde vários agentes estão responsáveis por resolver/cumprir subtarefas, e para tal comunicam numa linguagem de alto-nível. Existem ainda outras motivações associadas a esta área da Inteligência Artificial, nomeadamente:

- Problemas de grande dimensão serem resolvidos somente com a intervenção de um único agente;
- Interface simplificada para a cooperação entre homem e máquina, onde ambas as partes entram como agentes do sistema;

- Problemas geográfica e/ou funcionalmente distribuídos ganham uma solução natural;
- Existem problemas que o são de forma ainda mais acentuada porque os recursos tais como informações, conhecimento, peritos, entre outros, se encontram dispersos. Esta abordagem permite soluções muito mais eficientes tirando partido da distribuição de elementos existentes, com vista a resolver um dado problema;
- Aumento da rentabilidade dos recursos existentes para resolver um determinado problema, sendo este distribuído;
- Os SMAs são ideais quando existe uma distribuição espacial dos intervenientes de forma natural.

Portanto, num SMA, como já referido, existem vários agentes no mesmo ambiente, a trabalhar em conjunto no sentido de cumprir um determinado objetivo, ou cada um com o seu propósito, sendo que no entanto de uma forma global estão relacionados. Esta cooperação está subjacente nas aplicações de gestão de redes.

Podemos ter um SMA responsável pela gestão de uma rede onde os diferentes agentes podem ter responsabilidades e funções distintas.

A comunicação entre entidades computacionais sempre foi considerada um dos aspetos mais importantes na área da computação. No entanto, nos SMA a comunicação é tratada a um nível muito mais elevado quando comparável com outras áreas da computação, tratando-se de linguagens próximas das usadas pelos humanos devido à existência de todas as características que já foram referidas, nomeadamente as suas habilidades sociais [32, 39-42].

No âmbito deste trabalho um agente é visto como uma entidade que possui capacidades de perceção, processamento e atuação num determinado ambiente em que está inserido. Possui também conhecimento e a habilidade de raciocinar com base no seu conhecimento, para que possa tomar decisões em cada instante relativamente à melhor ação a desempenhar. Assumimos ainda que um agente é capaz de comunicar com outros agentes do mesmo ambiente ou mesmo com humanos, fazendo uso da sua habilidade social. Finalmente, possui mobilidade e inteligência [32, 39-42].

Para que um agente tenha a capacidade de comunicar é normalmente inserido um módulo de comunicação na sua arquitetura que se subdivide em duas componentes, mais concretamente: receção de mensagens e envio de mensagens. Este módulo está diretamente ligado ao módulo de inteligência, permitindo assim que este último módulo possa aceder às mensagens recebidas e determinar que mensagens irá enviar. De seguida é apresentada uma figura, figura 16, onde é apresentado o esquema genérico de um agente com habilidade de comunicação.

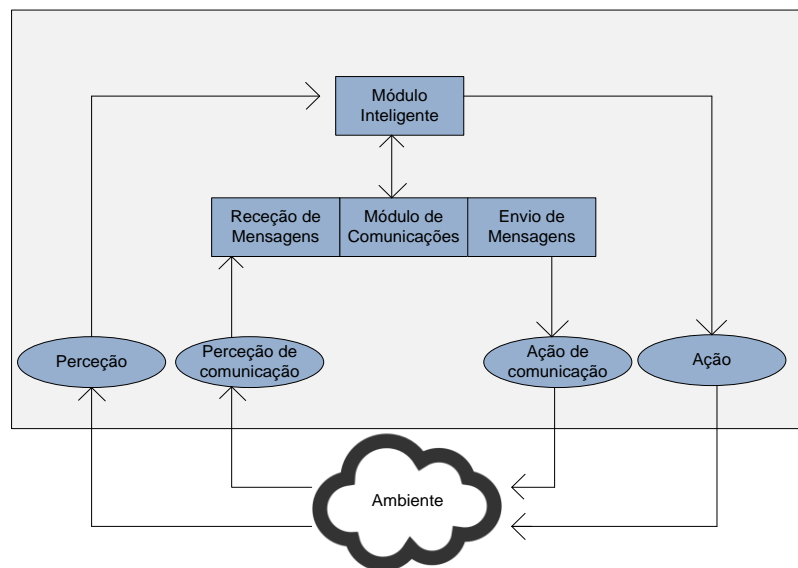


Figura 16 - Comunicação num Agente

É igualmente importante fazer referência a duas arquiteturas básicas do subsistema de comunicação entre os agentes. As arquiteturas são:

- **Comunicação Direta**, cada agente é responsável pela sua própria comunicação. Partilham especificações, enviando aos outros agentes as suas capacidades e/ou necessidades para que cada agente tome as suas decisões relativamente à comunicação. Os agentes comunicam diretamente entre si, não havendo a interferência de qualquer intermediário. A grande desvantagem desta arquitetura consiste no facto de não existir quem coordene a comunicação, o que pode levar a que o sistema bloqueie caso todos os agentes decidam enviar mensagens em simultâneo. A figura 17 ilustra esta arquitetura.

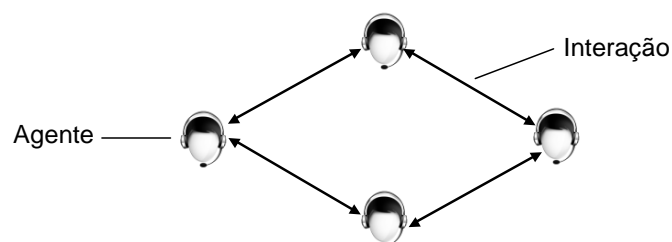


Figura 17- Comunicação Direta (baseada em [42])

- **Comunicação Assistida**, para os agentes comunicarem entre si, utilizam os serviços de outros agentes designados de "agentes Facilitadores". Ou seja, se um determinado agente quiser enviar uma mensagem para um outro agente, terá primeiro de enviar a mensagem para o agente Facilitador, que irá posteriormente enviá-la ao agente destinatário. Portanto, verifica-se aqui a existência de intermediário, deixando os agentes de comunicar diretamente entre si. Nesta arquitetura já não se verifica o problema da falta de coordenação que existe na arquitetura anterior uma vez que os

agentes não precisam de conter qualquer informação sobre os outros agentes, nem o seu endereço para que possam comunicar. Essa responsabilidade cabe ao agente Facilitador. Cada agente apenas tem de enviar a mensagem ao Facilitador, que tratará do resto. No entanto, surgem ainda assim alguns problemas com esta arquitetura, pois, o facto de existir um agente Facilitador pode levar a uma certa centralização no sistema e *bottleneck* no sistema de comunicação. Se este agente deixar de funcionar, deixa de haver comunicação entre os agentes. Na figura seguinte, figura 18, é apresentado um esquema que representa este tipo de comunicação [32, 39-42].

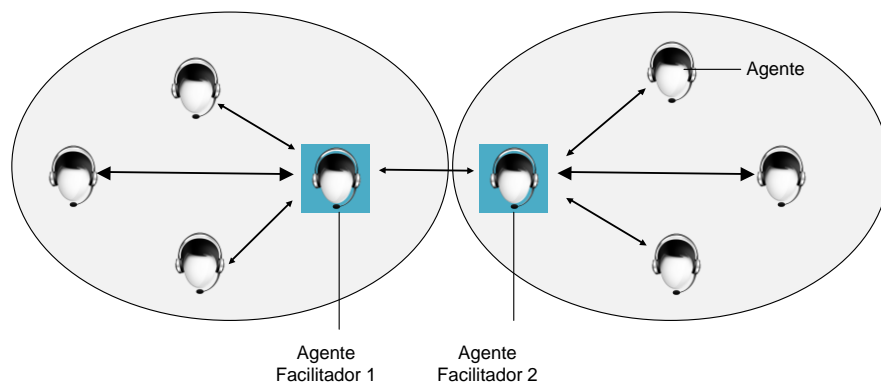


Figura 18- Comunicação Assistida (baseada em [42])

Portanto, podemos concluir que a utilização de um SMA nos oferece várias vantagens, nomeadamente:

- Paralelismo, permitindo que sejam atribuídas diferentes tarefas a cada um dos agentes. Assim, verifica-se a execução das mesmas num período de tempo mais reduzido;
- Maior fiabilidade, uma vez que não existe somente um ponto de falha (são utilizados vários agentes);
- Simplicidade, pois, verifica-se uma divisão de tarefas em subtarefas;
- Volume de comunicação mais reduzido, sendo esta feita a alto-nível como foi referido anteriormente, uma vez que o processamento está junto da fonte de informação [32, 39-42].

3.4 Agentes Móveis

Nos últimos anos o elevado crescimento de pessoas e dispositivos com acesso à *Internet*, bem como a enorme quantidade de serviços e informação que aí são disponibilizados, levaram a um elevado desenvolvimento nas tecnologias de computação distribuída e de rede [1, 43]. Este desenvolvimento levou à necessidade da criação de aplicações com a capacidade de trocar informação entre redes heterogéneas de forma eficiente, tendo sido assim criados vários paradigmas de *software* para computação distribuída. Aqui, podemos salientar os paradigmas Cliente-Servidor, *Remote Procedure Calling* (RPC) e agentes móveis.

3.4.1 Características dos Agentes Móveis

Agentes móveis consistem em processos de *software* com a capacidade de se mover de forma autónoma de um local de rede física para outro local a qualquer momento [43]. O estado de execução do processo é armazenado e movido para a nova localização com o código do agente permitindo que a execução possa ser retomada no novo *host* [43]. O agente executa a sua ação onde e quando bem entender e não precisa de estar instalado de forma prévia no cliente. Assim, é exibida a capacidade de autonomia inerente à mobilidade e execução do agente [43].

Os agentes móveis são uma forma específica dos paradigmas *mobile code* e *software agents*. Contudo, os agentes móveis são ativos relativamente à sua decisão de se mover entre os elementos da rede a qualquer altura na sua execução, ao contrário do que se passa com os outros paradigmas. Esta característica faz com que os agentes móveis sejam vistos como um solução poderosa e viável na implementação de aplicações distribuídas numa rede de computadores [43]. São um novo paradigma para os sistemas distribuídos, sendo possível identificar três características fundamentais [12,22,32,44-47]:

- Os agentes móveis estão por norma inseridos em ambientes de grande amplitude com redes heterogéneas e onde não existem garantias nem de fiabilidade nem de segurança relativamente a equipamentos e ligações de rede;
- Os agentes movem-se para aceder a recursos que estão disponíveis noutra local da rede, e não somente para balanceamento da carga como se verifica nos sistemas de objetos móveis;
- Os agentes móveis podem-se mover mais do que uma vez, sendo esta capacidade muitas vezes apelidada de *multi-hop*. Após se deslocar a um determinado serviço, o agente pode avançar para outros *hosts*, continuando a realizar as suas tarefas. Já nos paradigmas *Remote Evaluation* e *Code on Demand* o código móvel é apenas transferido uma vez.

3.4.2 Arquitetura Cliente-Servidor vs Agentes Móveis

A arquitetura cliente-servidor é o paradigma mais utilizado na computação distribuída. Esta arquitetura funciona à base de um pedido, ao qual é dada uma resposta. Todos os elementos são estacionários relativamente à execução, um cliente ao necessitar de um determinado serviço que é disponibilizado por um determinado servidor, efetua em pedido (que normalmente contém o nome do serviço requisitado e mais alguns parâmetros), esse pedido é analisado pelo servidor e devolvida uma resposta que consiste na informação solicitada pelo cliente. Nos últimos anos tem-se verificado uma crescente utilização dos *Web Services*, que permitem a utilização deste paradigma à escala da *Internet*. Na próxima figura, figura 19, é apresentado um exemplo deste tipo de implementação, sendo o servidor o *Componente B* e o cliente o *Componente A* [12, 32, 48-51].

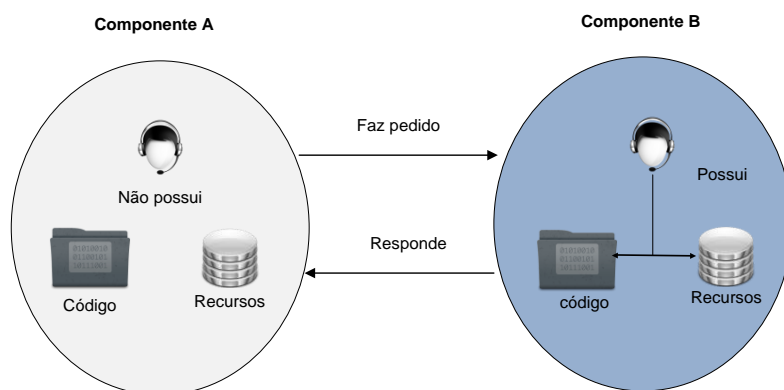


Figura 19- Arquitetura Cliente-Servidor

No caso de uma arquitetura onde existam agentes móveis, tem de ser vista de forma distinta da anterior, surgindo assim como um novo paradigma. De seguida é apresentado um exemplo (figura 20 [12, 48-51]) que serve como base de implementação deste novo paradigma onde são inseridos agentes móveis.

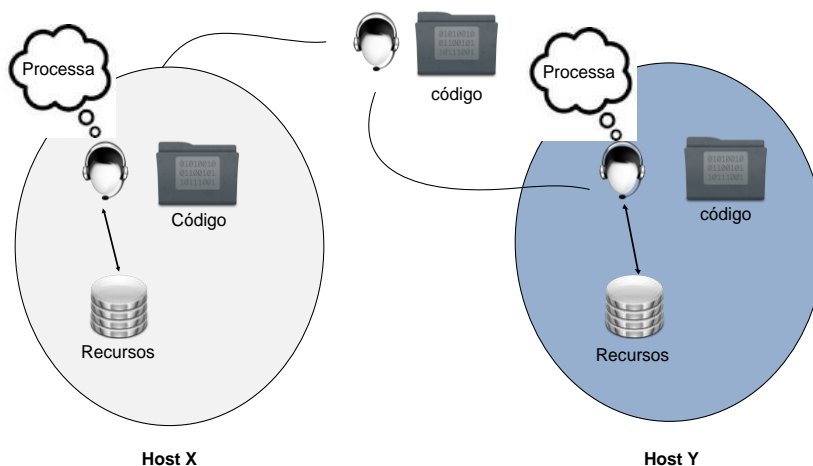


Figura 20- Arquitetura Agentes Móveis

O *Host X* contém código e acede aos recursos que necessita para executar a(s) sua(s) tarefa(s). Posteriormente, move-se para o *Host Y* devido ao facto de necessitar de outros/mais recursos, continuando a sua execução. O agente não se move apenas quando necessita de aceder a outros recursos, poderá também mover-se devido ao simples facto de ter terminado as tarefas que lhe foram atribuídas inicialmente, estando portanto disponível para migrar para outro *host* da rede que precise das suas funcionalidades, quer para o mesmo tipo de tarefas, quer para a realização de outras [12, 48-51].

Se o agente for dotado de inteligência, tem a capacidade de tomar decisões e consequentemente agir, em função da sua perceção relativamente ao ambiente em que está inserido e dos eventuais acontecimentos que surjam à medida que a sua execução avança. Tanto pode decidir terminar uma tarefa, como adaptar-se para que possa cumpri-la de forma diferente do habitual, caso surjam problemas, ou até migrar para um novo destino [12, 48-51].

3.4.3 Vantagens na Utilização de Agentes Móveis

Neste trabalho os agentes móveis são utilizados para ultrapassar alguns dos problemas existentes nas abordagens tradicionais de redes e já por nós identificados e apresentadas no capítulo 2.

Assim, do estudo efetuado foram identificadas as principais características dos agentes móveis que os tornam especialmente adequados para utilizar em aplicações de gestão de redes [3, 12, 13, 52-60]:

- Os agentes móveis conseguem guardar determinados parâmetros, continuando posteriormente a sua execução em outro *host*, onde as várias tarefas vão ocorrer localmente e não com o auxílio de recursos da rede. Este procedimento diminui o número de ligações que são necessárias estabelecer para realizar as tarefas. Para além disso, o volume de dados a transferir diminui consideravelmente, pois, o processamento é feito localmente. A próxima figura, figura 21, ilustra estes aspetos;

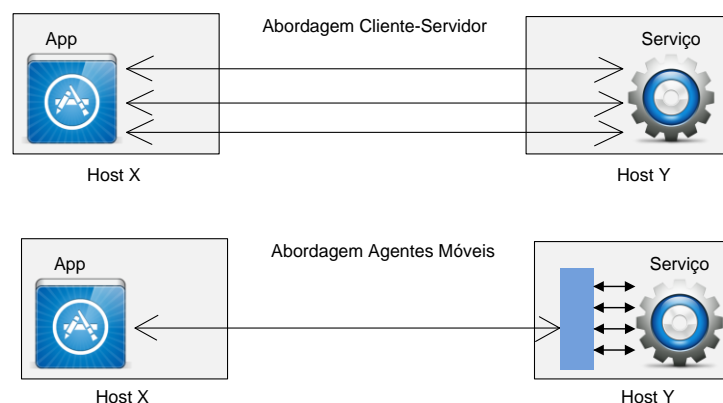


Figura 21- Abordagem Cliente-Servidor vs Agentes Móveis

- Próximo da ausência de latências na rede, tratando-se de um sistema distribuído, e adicionando o facto de na maioria dos casos serem necessárias respostas imediatas, quando elevamos o número de comunicações é natural que surjam latências na rede. Nas soluções baseadas em agentes móveis é possível enviá-lo para os pontos da rede onde é necessário efetuar tarefas de gestão. Assim, as latências de rede são reduzidas quase a zero;
- Encapsulamento de protocolos, na existência das inerentes trocas de informação num sistema distribuído, é necessário que cada *host* esteja preparado para interpretar os dados recebidos e enviados, estando dotado de protocolos específicos para tal. No entanto, esses protocolos vão evoluindo, passando a incluir novas funcionalidades, alteram-se e em cada *host* passa a ser necessário atualizá-los. Essa tarefa é muito difícil e trabalhosa para os administradores de sistemas, que muitas vezes não a concretizam. Os agentes móveis, por seu lado, conseguem mover-se para qualquer local, estabelecendo os seus próprios canais de comunicação baseados num protocolo implementado nos mesmos;
- Execução assíncrona e autónoma, os agentes móveis podem ser configurados para levarem a cabo um conjunto de tarefas num determinado local para onde os agentes são deslocados. Dessa forma, estamos a dar-lhe autonomia e estes podem operar assincronamente, independentes do processo que os criou originalmente. O agente móvel pode voltar a conectar-se mais tarde, após completar a execução das tarefas, ou numa altura que considere ideal, de acordo com as suas capacidades autónomas. A figura seguinte, figura 22, ilustra esta capacidade;

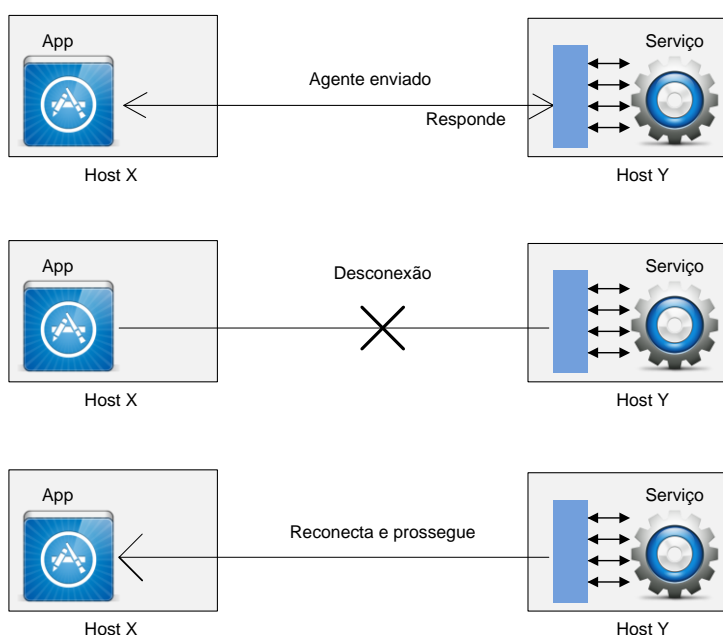


Figura 22- Execução Assíncrona

- Dinamismo, os agentes móveis podem fazer uso da sua autonomia e pró atividade, podendo tomar decisões com base nas informações (percepções) que vão obtendo do ambiente em que estão inseridos. Se um determinado problema surgir, vários agentes podem ainda distribuir-se de forma inteligente, conseguindo resolver um determinado problema que necessite da sua presença em vários *hosts*, ou seja, através de uma solução distribuída;
- Heterogeneidade, relativamente à computação em rede, esta é naturalmente heterogénea, quer a nível do *hardware*, quer a nível do *software*. Os agentes móveis são independentes dos computadores e camadas de transporte, dependendo apenas dos seus ambientes de execução. Consegue-se assim obter condições favoráveis para integrar redes heterogéneas;
- Robustez e tolerância a falhas, o dinamismo associado aos agentes móveis permite que estes reajam a situações adversas de forma autónoma, permitindo obter um sistema robusto e tolerante a falhas. Situações como por exemplo, um *host* que vá encerrar e onde existam agentes a executar, é possível informar os agentes, guardar o seu estado e migrar para um outro local, continuando a sua execução.

3.4.4 Conclusão

Neste capítulo foi feito o estudo relativamente aos agentes móveis. O estudo focou-se em áreas como a tolerância a falhas, configuração, segurança, entre outras. No entanto, é de salientar que o paradigma agentes móveis permite igualmente atingir objetivos relacionados com as todas as áreas funcionais da gestão de redes. Atualmente, os agentes móveis continuam a ser tema de investigação e desenvolvimento, apesar de não serem muito utilizados em redes/aplicações de grande escala. Esta realidade está diretamente relacionada com o facto de não existirem aplicações comerciais baseadas neste paradigma.

Outro motivo prende-se com a necessidade de desenvolver este tipo de sistemas para que todas as plataformas o possam utilizar, criando ambientes em que haja interoperabilidade. Pretende-se que não se note diferenças ou exclusões entre sistemas operativos, fabricantes de ativos de rede ou mesmo aplicações de *software*. Contudo, em algumas áreas como as redes ativas, tem-se verificado a utilização, ou a tentativa de utilizar, conceitos de agentes móveis, como por exemplo, pacotes que circulam na rede carregarem código que será executado em *switches*, permitindo a sua configuração dinamicamente. Outras aplicações têm surgido na área das telecomunicações e mesmo na área do comércio eletrónico, e-commerce [32].

4 Simulador de Redes

4.1 Introdução

Neste capítulo são introduzidas as ferramentas de simulação de redes e apresentadas algumas das ferramentas que se encontram, atualmente, disponíveis no mercado. Explica-se o seu funcionamento, características, e é ilustrada a forma como podem contribuir para a gestão de redes. Por último é apresentada a ferramenta de simulação utilizada neste trabalho, assim como as razões que nos levaram a efetuar tal escolha. São ainda feitas algumas considerações finais.

4.2 Ferramentas de Simulação – Principais características e funcionalidades

Iremos apresentar três ferramentas de simulação de redes que se encontram disponíveis no mercado, nomeadamente as ferramentas *Cisco Packet Tracer*, *PSimulator 2* e a *Graphical Network Simulator 3*.

4.2.1 *Cisco Packet Tracer*

Esta ferramenta, como o próprio nome indica, foi desenvolvida por um dos maiores fabricantes de dispositivos e soluções de redes, a *Cisco Systems*, com o intuito de ser utilizada pelo fabricante nas suas academias de formação, conhecidas mundialmente como *Cisco Networking Academy*. Esta ferramenta permite ao utilizador:

- A visualização do ambiente de rede para a criação, configuração e solução de problemas;

Simulador de Redes

- A criação e visualização da transmissão de pacotes de dados virtuais da rede criada em tempo real;
- Encontrar soluções para eventuais problemas que possam surgir;
- Criar e configurar complexas topologias de redes com dispositivos virtuais e conexões simuladas, que numa situação real iriam exigir grande quantidade de recursos.

Para que o utilizador possa desenvolver as suas atividades é disponibilizada uma interface *drag-and-drop* bastante didática e simples. Aliás, uma das grandes qualidades desta ferramenta, para além de todas as suas funcionalidades, é a sua interface bastante simples, prática e eficiente. Na imagem que apresentamos de seguida, figura 23, é possível visualizar a interface gráfica desta ferramenta.

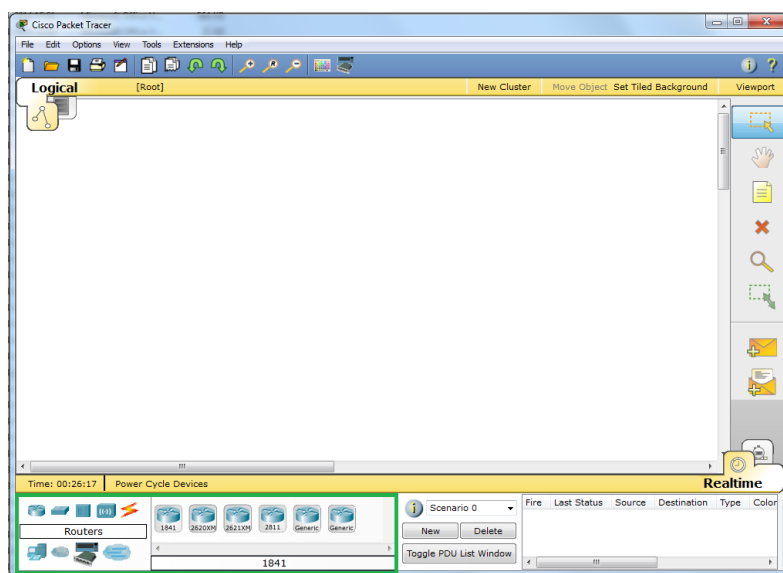


Figura 23 - Interface do simulador *Cisco Packet Tracer*

Relativamente à construção das topologias de rede o processo é bastante fácil e simples. A ferramenta dispõe de um vasto conjunto de dispositivos de rede (figura 24) que juntamente com uma funcionalidade do tipo *drag-and-drop* permite que o utilizador possa construir facilmente o cenário a desenvolver.

Ferramentas de Simulação – Principais características e funcionalidades

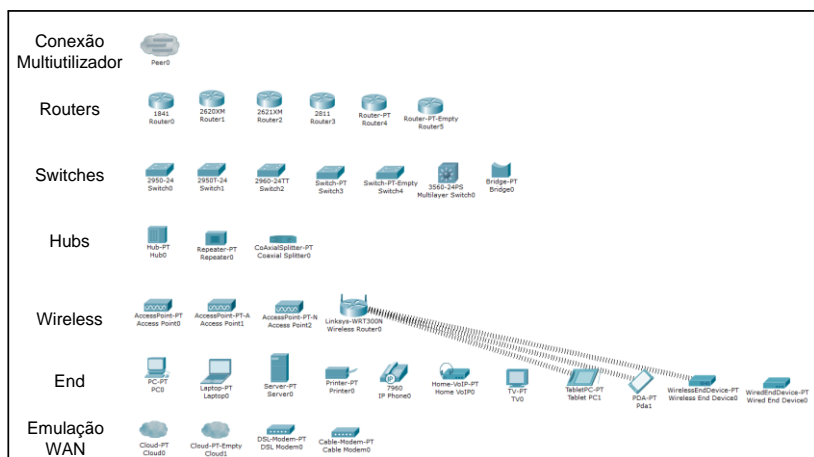


Figura 24- Dispositivos suportados pelo simulador

O processo de configuração de dispositivos de rede é igualmente simples. Para cada dispositivo, a ferramenta apresenta uma janela onde se encontram as opções de configuração suportadas para o dispositivo em causa, onde existe também a possibilidade do utilizador acrescentar módulos de *hardware* ao dispositivo de rede, conforme ilustrado na figura 25. Tal possibilidade permite-nos concluir que a ferramenta suporta dispositivos modulares. A ferramenta permite ainda duas formas de configuração: uma mais rápida e simples, onde são inseridas as configurações mais básicas na interface gráfica, ou por linha de comandos como se estivéssemos a aceder por consola. Estas duas opções tornam-se extremamente úteis quando o utilizador pretende usar ativos de rede como *switches* ou *routers*. O utilizador pode não ter conhecimentos suficientes para utilizar comandos de configuração para os ativos de rede que pretende usar, não ficando assim impedido, ou limitado, para construir os seus cenários. As figuras seguintes, figura 25 e 26, permitem visualizar esta funcionalidade, sendo utilizada a configuração de um *router* como exemplo.

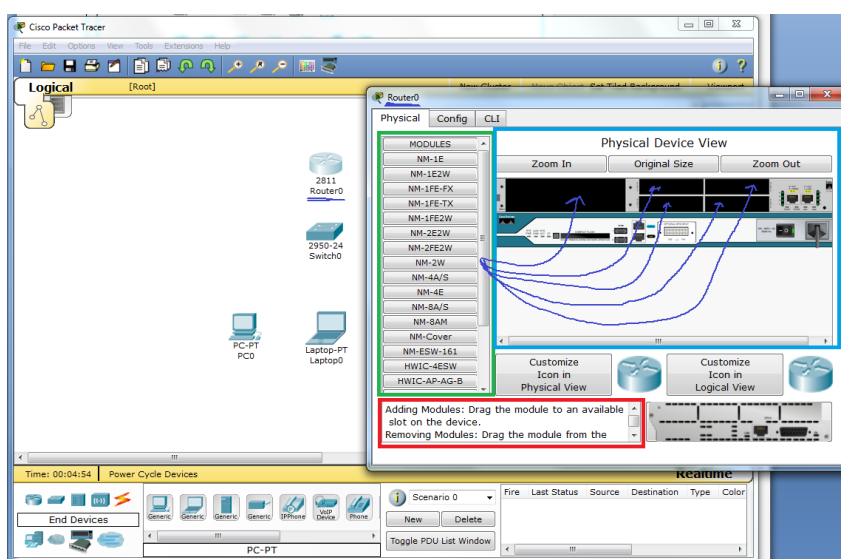


Figura 25- Exemplo de adição de módulos num *router*

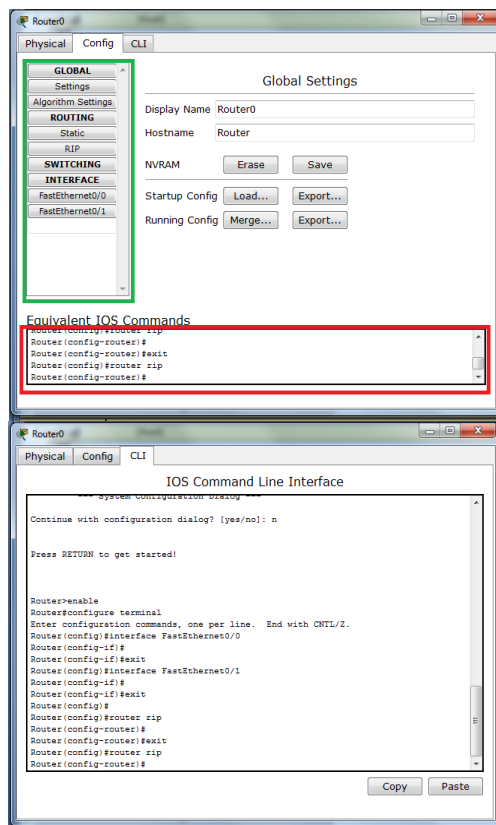


Figura 26- Opções de Configuração

Outra funcionalidade que o *Packet Tracer* disponibiliza, consiste numa *tag "Desktop"* que é apresentada para alguns *end devices* como computadores, *smartphones*, *tablets*, etc. Nesta *tag* são apresentadas ao utilizador opções de configuração e funcionalidades, como criar VPNs, configurar contas de *email*, utilizar um *Web Browser*, *Command Prompt*, entre outros. Na figura seguinte, figura 27, é possível verificar estas opções.

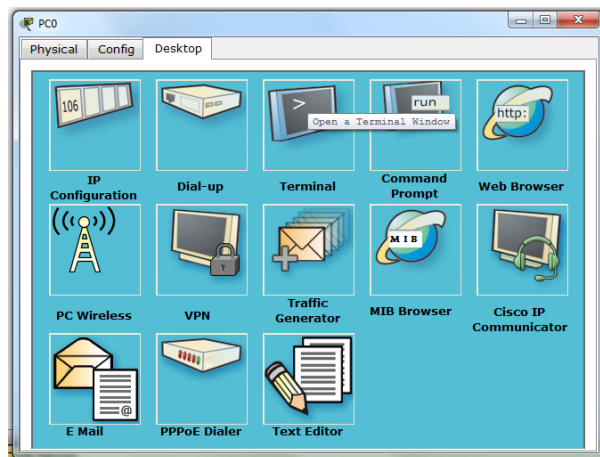


Figura 27- "Tag" Desktop

No que diz respeito à conexão dos dispositivos, esta ferramenta apresenta vários tipos de conexão, bem como a possibilidade da ferramenta escolher automaticamente o tipo de conexão que deverá ser usada para conectar dois dispositivos, facilitando assim, a escolha o utilizador em caso de dúvida ou desconhecimento. Essas conexões podem ser do tipo: consola; cabo direto; cabo cruzado; série DCE; série DTE; fibra; telefone e coaxial. Na figura 28 apresentamos estas opções na interface do *Cisco Packet Tracer*.

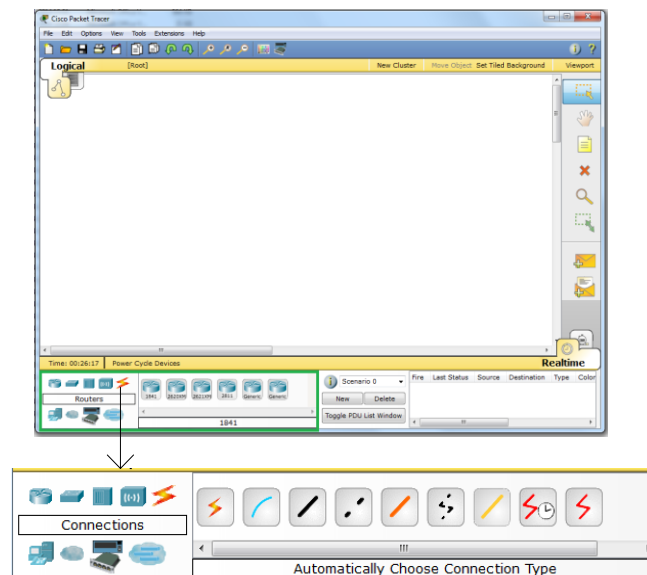


Figura 28- Tipos de conexão no *Packet Tracer*

A ferramenta contém duas áreas de trabalho distintas, uma física e outra lógica. A parte lógica permite aos utilizadores criar as suas topologias de rede, onde adicionam os dispositivos que pretendem usar, efetuam as configurações necessárias e realizam as conexões entre os dispositivos. Ou seja, é onde estão inseridas as funcionalidades referidas por nós. Relativamente à parte física, consiste numa dimensão física em modo gráfico. Isto é, é dada ao utilizador a noção de como os dispositivos de rede ficariam num ambiente real. Por exemplo, se o utilizador criar uma topologia com um *router*, um *switch* e um computador (na parte lógica), na parte física, o simulador apresenta esses equipamentos ao utilizador, como se este estivesse fisicamente na presença dos mesmos. Esta área de trabalho suporta ainda representações gráficas de redes, edifícios, bastidores, entre outras. Na figura 29 é possível ver a diferença entre ambas as áreas de trabalho, utilizando um cenário construído com três *routers*, um *switch*, um servidor, um computador e ainda as conexões entre os dispositivos.

Simulador de Redes

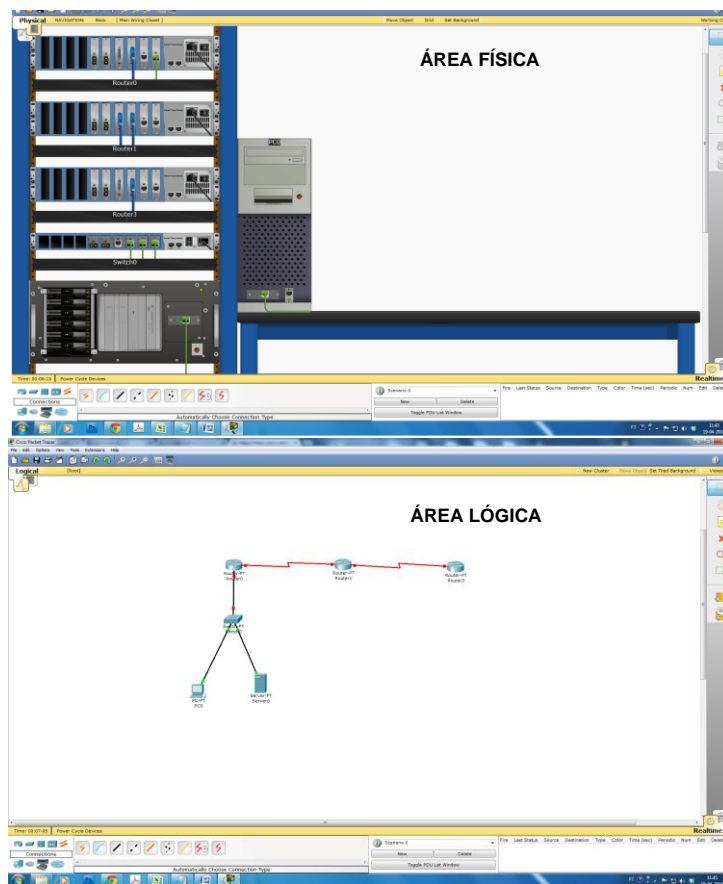


Figura 29- Área Lógica vs Área Física

É possível analisar e visualizar o comportamento da rede. Para tal, são disponibilizados dois modos: tempo real e simulação. No primeiro modo, a rede tem um comportamento tal e qual como teria se fosse uma rede real, isto é, os dispositivos funcionam e respondem em tempo real às atividades de rede. No modo simulação, o utilizador pode visualizar e controlar intervalos de tempo, funcionamento interno de transferência de dados e a propagação de dados ao longo da rede. Assim, é possível ao utilizador perceber melhor o funcionamento da rede.

No que a protocolos diz respeito, o *Packet Tracer* suporta um vasto conjunto de protocolos inseridos nas diferentes camadas do modelo TCP/IP. Na figura 30 estão ilustrados os principais protocolos que são suportados pela ferramenta.

Layer	Cisco Packet Tracer Supported Protocols
Application	<ul style="list-style-type: none"> FTP, SMTP, POP3, HTTP, TFTP, Telnet, SSH, DNS, DHCP, NTP, SNMP, AAA, ISR VOIP, SCCP config and calls, ISR command support, Call Manager Express
Transport	<ul style="list-style-type: none"> TCP and UDP, TCP Nagle Algorithm & IP Fragmentation, RTP
Network	<ul style="list-style-type: none"> BGP, IPv4, ICMP, ARP, IPv6, ICMPv6, IPsec, RIPv1/v2/ng, Multi-Area OSPF, EIGRP, Static Routing, Route Redistribution, Multilayer Switching, L3 QoS, NAT, CBAL, Zone-based policy firewall and Intrusion Protection System on the ISR, GRE VPN, IPsec VPN
Network Access/Interface	<ul style="list-style-type: none"> Ethernet (802.3), 802.11, HDLC, Frame Relay, PPP, PPPoE, STP, RSTP, VTP, DTP, CDP, 802.1q, PAgP, L2 QoS, SLARP, Simple WEP, WPA, EAP

Figura 30- Protocolos suportados pelo *Packet Tracer*

Existem ainda outras funcionalidades bastante interessantes que achamos melhor referir. A ferramenta contém um modo multiutilizador *peer-to-peer* que permite a construção colaborativo de redes virtuais sobre uma rede real. Verifica-se também a preocupação da Cisco em ajudar o utilizador em realizar as suas tarefas, uma vez que a ferramenta inclui informação e sugestões que podem ser seguidas pelo utilizador. Para além disso, o utilizador também pode aprofundar os seus conhecimentos através da resolução de atividades de aprendizagem personalizadas denominadas "*Activity Wizard*".

4.2.2 *Psimulator 2*

O *Psimulator 2* é um simulador de redes básico que foi desenvolvido na Universidade Técnica de Praga para ser utilizado no ensino de determinados tópicos de redes de computadores. Possui a particularidade de necessitar que o sistema onde é executado o simulador suporte a tecnologia Java. O *Psimulator 2* permite aos seus utilizadores criarem topologias de redes de uma forma simples e didática através de uma interface gráfica onde são disponibilizados diferentes dispositivos de rede, tipos de conexões e opções de configuração para cada dispositivo utilizado. A figura seguinte, figura 31, ilustra a interface gráfica do simulador.

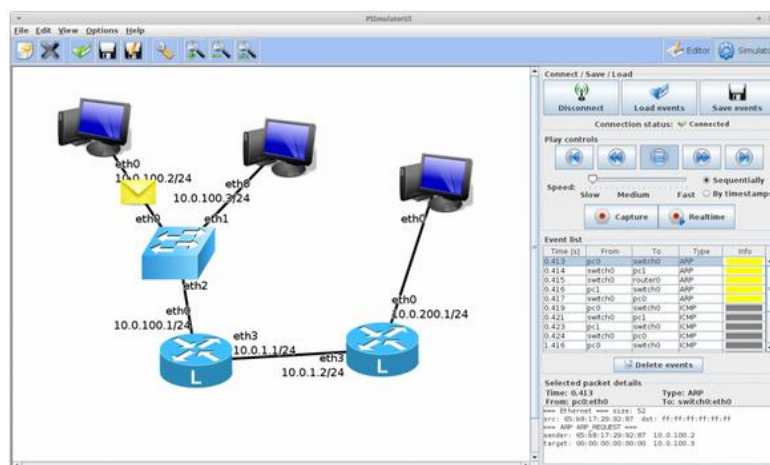


Figura 31- Interface gráfica do *PSimulator*

Relativamente a dispositivos disponibilizados para a construção de cenários, a ferramenta contém apenas três tipos: computadores, *switches* e *routers*, sendo cada um destes nós um programa em Java que implementa as funcionalidades do dispositivo. Uma vez iniciada a simulação da rede o utilizador pode conectar-se a cada dispositivo desse cenário através de *telnet* e configurá-los através de comandos específicos. Portanto, é também necessário que exista um cliente *telnet* para o simulador funcionar.

Relativamente aos recursos que esta ferramenta apresenta para a elaboração do cenário, o utilizador dispõe de quatro opções de dispositivos de rede que podem ser usados: PC, *Switch*, *Router* e um ao qual denominam de "*Real PC*". Para cada um dos dispositivos existem várias versões disponíveis, onde a diferença entre as versões está no número de interfaces que o dispositivo terá disponíveis, como ilustrado na figura 32 para o caso particular de um *router*.

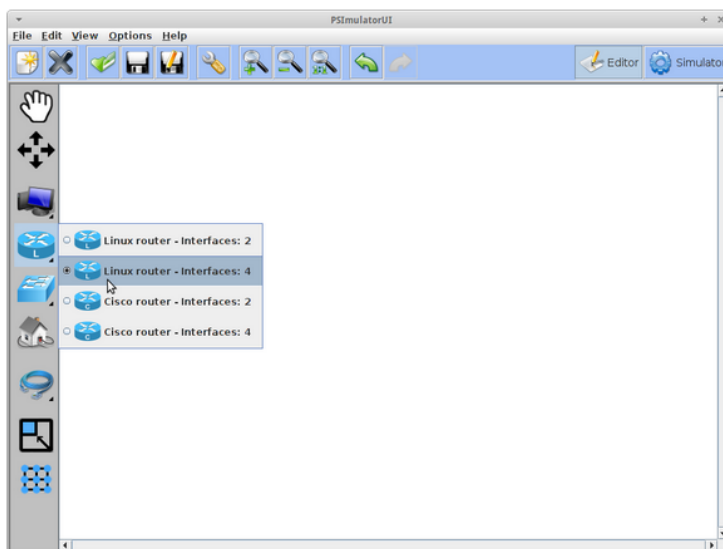


Figura 32 - Exemplo de escolha de um dispositivo no PSimulator

A forma como o utilizador poderá efetuar as configurações nos dispositivos é bastante simples. Tal como acontece com o *Cisco Packet Tracer* é disponibilizada uma janela de configuração para cada dispositivo, onde são apresentados ao utilizador as opções de configuração disponíveis para o dispositivo em causa. Estas configurações baseiam-se basicamente na adição/remoção de interfaces de rede, seu endereçamento, estado, entre outras. Na figura 33 apresentamos uma dessas janelas de configuração, neste caso de um *router*.

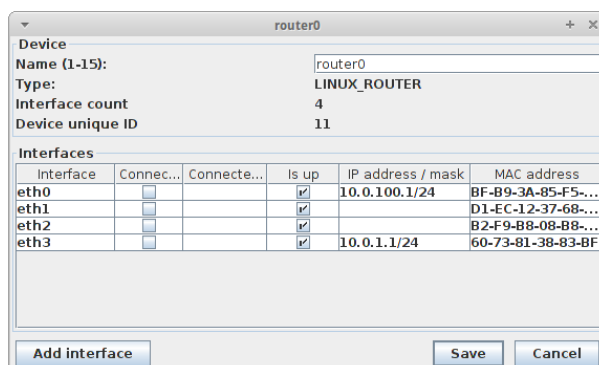


Figura 33 - Janela de configuração de um *router* no *PSimulator*

Ao contrário do que acontece com a ferramenta que apresentamos anteriormente em que eram disponibilizados vários tipos de conexão, o *PSimulator 2* disponibiliza apenas dois tipos de ligação *Ethernet*. A diferença entre ambas está no *delay*.

O *PSimulator* está dividido em duas partes: *front-end* e *back-end*, sendo que ambos trabalham em conjunto para que utilizador possa realizar as suas simulações. O *front-end* consiste na interface gráfica que é disponibilizada ao utilizador e que lhe permite construir de uma forma amigável os seus cenários, assim como verificar o tráfego de pacotes, pacotes enviados, recebidos, perdidos, etc., quando inicia a simulação do cenário construído. Relativamente ao *back-end* é o responsável por executar a simulação. O processo de simulação é um pouco complexo, uma vez que o utilizador necessita de executar um conjunto de passos para que a rede virtual esteja a funcionar corretamente. Uma vez construída a topologia, esta é armazenada num ficheiro XML que é criado pelo *front-end* e fornecido ao *back-end* quando este é iniciado pelo utilizador. De seguida, o utilizador necessita ainda de fazer conexão a um servidor inserido na ferramenta, denominado de *back-end server*. O *PSimulator* disponibiliza para cada nó uma janela de *telnet* onde o utilizador poderá efetuar configurações adicionais através de comandos específicos. Estas configurações consistem:

- Permitir IP Forwarding nos *routers*;
- Criação de rotas estáticas nos *routers*;
- Criar uma rota *default* em cada computador.

Uma vez configurada toda a rede a ferramenta permite então que o utilizador execute alguns testes a nível de tráfego, assim como capturar pacotes de dados. Estes testes são efetuados através da execução de comandos *ping* e *traceroute*, que são os únicos suportados por esta ferramenta para estes testes. Desta forma, é possível ao utilizador verificar a troca de pacotes de dados entre os nós da rede e analisá-los de uma forma mais pormenorizada, assim como, verificar qual o caminho que é seguido pelos pacotes desde a sua origem até ao nó de destino. Estas funcionalidades estão representadas nas figuras 34 e 35.

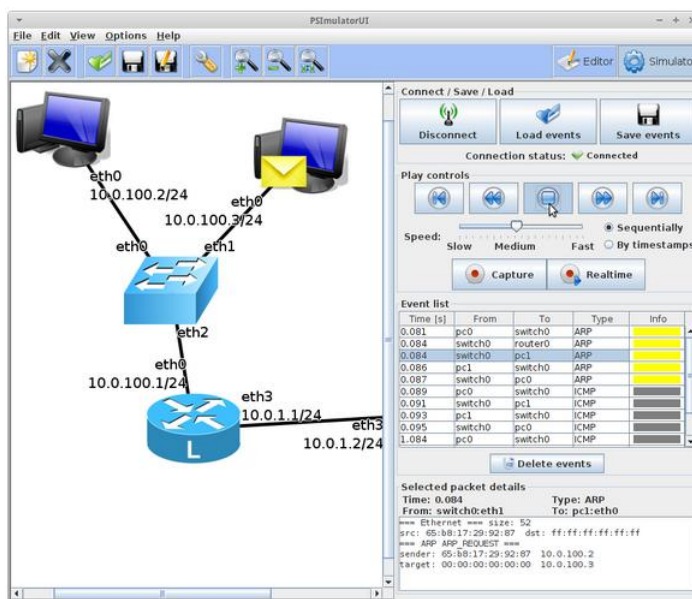


Figura 34 - Análise do tráfego de pacotes no PSimulator

```
# pc1:/#traceroute 10.0.200.2
# traceroute 10.0.200.2
traceroute to 10.0.200.2 (10.0.200.2), 30 hops max, 60 byte packets
 1 10.0.100.1 (10.0.100.1) 25.064 ms 15.272 ms 11.69 ms
 2 10.0.1.2 (10.0.1.2) 19.507 ms 28.932 ms 25.112 ms
 3 10.0.200.2 (10.0.200.2) 27.459 ms 28.578 ms 30.476 ms
#
```

Figura 35 - Resultado de um traceroute no PSimulator

4.2.3 Graphical Network Simulator - GNS 3

O GNS 3 é um simulador que permite ao utilizador realizar simulações de redes de dados complexas, à semelhança de programas como o VMWare, VirtualBox ou Virtual PC, que permitem simular vários sistemas operativos num ambiente virtual. No entanto, o GNS 3 permite o mesmo tipo de simulações mas utiliza sistemas operativos do fabricante Cisco, possibilitando ao utilizador a execução de um Cisco IOS num ambiente virtual, no seu computador pessoal (independentemente do sistema operativo). Para tal ser possível, o GNS 3 é executado por cima de um outro programa denominado Dynamips, que para além de disponibilizar uma interface simples, atrativa e didática ao utilizador, é quem permitem estas emulações dos sistemas operativos Cisco.

Esta ferramenta também suporta outros programas de emulação como Qemu, Pemu e VirtualBox, que permitem que sejam utilizadas firewalls Cisco ASA e PIX, Cisco IPS, routers Juniper e hosts com sistemas operativos Linux, Windows, Mac OS X, etc. Isto faz com que o utilizador possa por exemplo, ter um router Cisco a comunicar com um host que utiliza Windows. Outra particularidade desta ferramenta está relacionada com a simulação de routers. Ao contrário do que acontece em muitos simuladores em que existem sempre

comandos ou determinados parâmetros que não são suportados, no GNS 3 o sistema operativo utilizado no *router* virtual é o mesmo que o utilizado no dispositivo real. Ou seja, os comandos e parâmetros suportados são exatamente os mesmos que o IOS produz e suporta, mesmo tratando-se de um dispositivo virtual. Tem ainda como vantagem o facto de ser um *software Open Source*, sendo a sua utilização completamente livre e grátis. No entanto, apresenta algumas restrições a nível de licenciamento, ficando ao encargo do utilizador obter os sistemas operativos com que pretende trabalhar e fornecê-los à ferramenta. Ou seja, a ferramenta suporta o dispositivo, mas para ser usado no cenário virtual, o utilizador tem de fornecer o sistema operativo.

O GNS 3 também possibilita a simulação de dispositivos como *Switchs Ethernet, Frame-Relay e ATM*. No entanto, a sua simulação é superficial, isto é, são abrangidas apenas as características básicas de cada equipamento.

Esta ferramenta detém a capacidade de comunicar com uma rede real, permitindo testar cenários reais. Existe igualmente a possibilidade de conectar equipamentos reais (*routers, switchs, etc.*) a um cenário simulado. Estas funcionalidades trazem várias vantagens numa rede, uma vez que é possível ao administrador efetuar testes num determinado equipamento antes de este ser colocado em produção, simular novos cenários para uma rede que já se encontra a funcionar, testar novos sistemas operativos, etc., garantindo assim que as alterações não irão colocar em causa o funcionamento normal da rede como um todo. Portanto, o GNS 3 diminui bastante a distância entre um cenário virtual e real, uma vez que são utilizados sistemas operativos nos dispositivos virtuais tal e qual como são usados nos reais. É possível utilizar máquinas virtuais com o sistema operativo e com as configurações desejadas. O GNS 3 possibilita ainda interligar o cenário virtual a uma rede real.

Como referido anteriormente, o GNS 3 apresenta uma interface bastante intuitiva onde o utilizador pode construir as suas topologias de rede. Esta interface gráfica funciona do modo *drag-and-drop*, e é apresentada na figura seguinte, figura 36.

Simulador de Redes

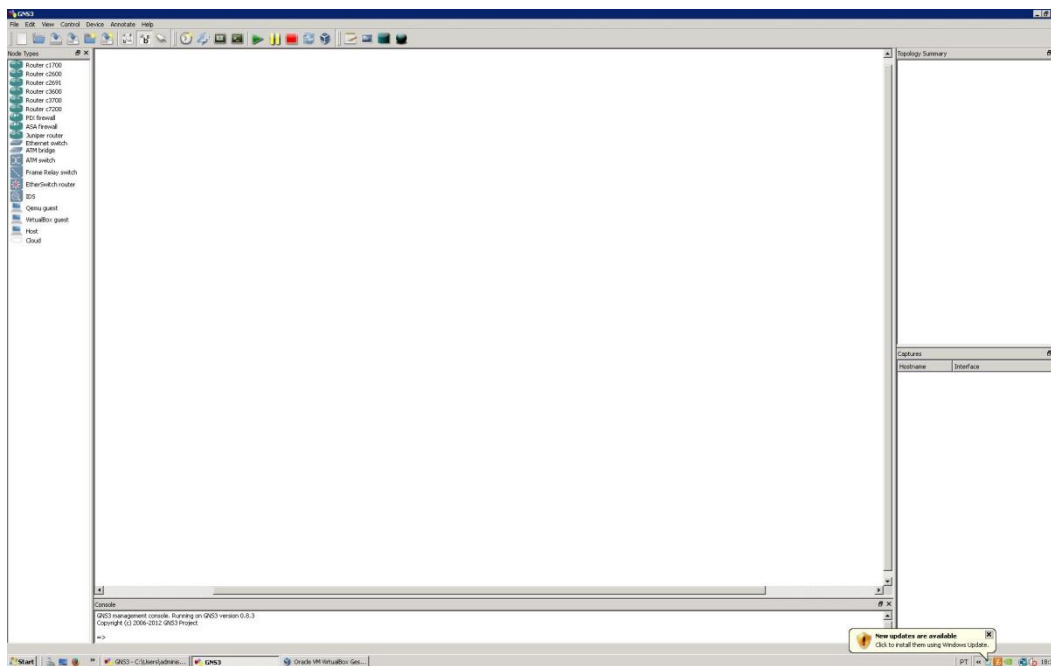


Figura 36 - Interface gráfica do GNS 3 e principais blocos

Na interface é disponibilizado ao utilizador um conjunto vasto de dispositivos de rede que são suportados pelo simulador e que podem ser inseridos pelo utilizador nos seus cenários. Estes dispositivos variam entre *routers*, computadores (*hosts*), *switches ethernet*, *firewalls*, máquinas virtuais e *cloud*. Na figura 37 é possível visualizar estes dispositivos na interface do GNS 3.

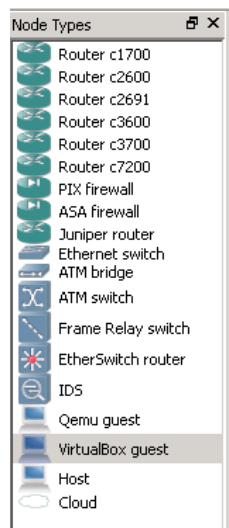


Figura 37 - Dispositivos disponíveis no GNS3

Dispositivos como *routers* e *switches* necessitam de um conjunto de procedimentos e configurações que o utilizador necessita fazer antes de os usar. Como foi referido anteriormente, os dispositivos que estão inseridos no GNS3 não suportam sistemas operativos. Para que o utilizador possa utilizar dispositivos como *routers* e *switches* da *Cisco* ou da *Juniper*

tem primeiro que fornecer à ferramenta os sistemas operativos referentes aos dispositivos que pretende usar. O mesmo acontece com as máquinas virtuais, em que para serem utilizadas têm de estar previamente criadas no *VirtualBox* e garantida a existência de uma ligação entre ambos os programas. No entanto, este processo só precisa de ser efetuado na primeira vez que o utilizador usar o *Virtual Box* no GNS 3.

Outro aspeto importante referente a esta ferramenta está relacionado com a simulação de computadores (*hosts*). Ao contrário do que acontece no *Packet Tracer* em que é suportada a simulação do "desktop" de um computador, no GNS 3 tal não acontece. O GNS 3 recorre a um programa externo chamado VPCS (*Virtual PC Simulator*). Os computadores que são adicionados no cenário são configurados no VPCS, sendo posteriormente feita a ligação entre os dois programas. O que é fornecido pelo VPCS no fundo consiste na simulação de um *command prompt* do computador que se encontra no cenário do GNS 3, permitindo assim que o utilizador possa realizar certas tarefas como por exemplo testar por linha de comandos a comunicação entre diferentes computadores em diferentes redes, executando comandos *ping*, tal como ilustrado na figura 38.

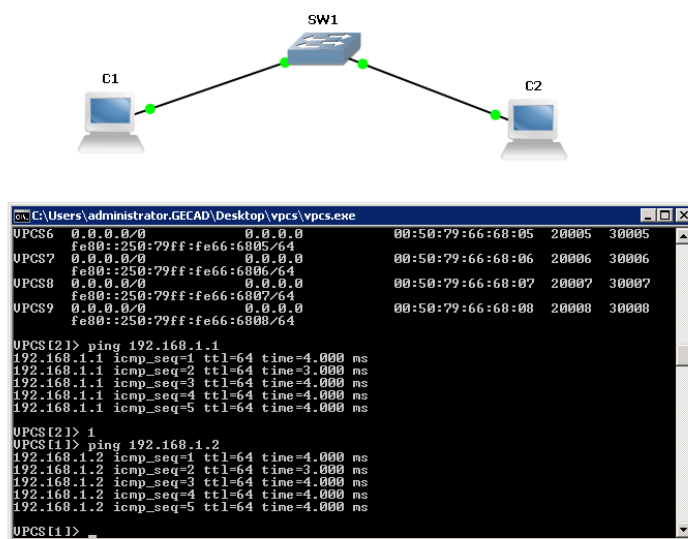


Figura 38 - Interface do VPCS

Relativamente a tipos de conexões entre os dispositivos de rede, esta ferramenta apresenta várias opções que podem ser usadas pelo utilizador, mais especificamente: *GigaEthernet*, *ATM*, *Manual*, *POS (Packet Over SONET)*, *FastEthernet*, *Ethernet* e *Serial*.

No que se refere à configuração dos dispositivos, nos vários dispositivos suportados pelo GNS 3, esta é feita de uma forma bastante simples. Tal como acontece nos simuladores *Packet Tracer* e *PSimulator 2*, o GNS 3 suporta para cada dispositivo uma janela de configuração onde são apresentadas as opções de configuração que o dispositivo selecionado suporta. Essas opções vão desde a criação e configuração de VLAN em *switches*, ao número de interfaces de

rede e seus modelos no caso das máquinas virtuais, até à adição de *slots* de *hardware* num *router*, à semelhança do que acontece no *Packet Tracer*.

De salientar que no caso específico dos *routers*, a janela de configuração apresenta apenas opções relacionadas com *hardware*, como por exemplo configuração de memória, discos e adição de *slots*. Isto deve-se ao facto de o *router*, apesar de virtual, conter um IOS real que é atribuído pelo utilizador tal como referido anteriormente. Portanto, as restantes configurações são feitas diretamente no IOS como acontece com um dispositivo real, através de comandos de configuração do fabricante. Na figura seguinte, figura 39, apresenta-se um exemplo de uma janela de configuração, neste caso de um *router*, assim como a sua janela de configuração do IOS.

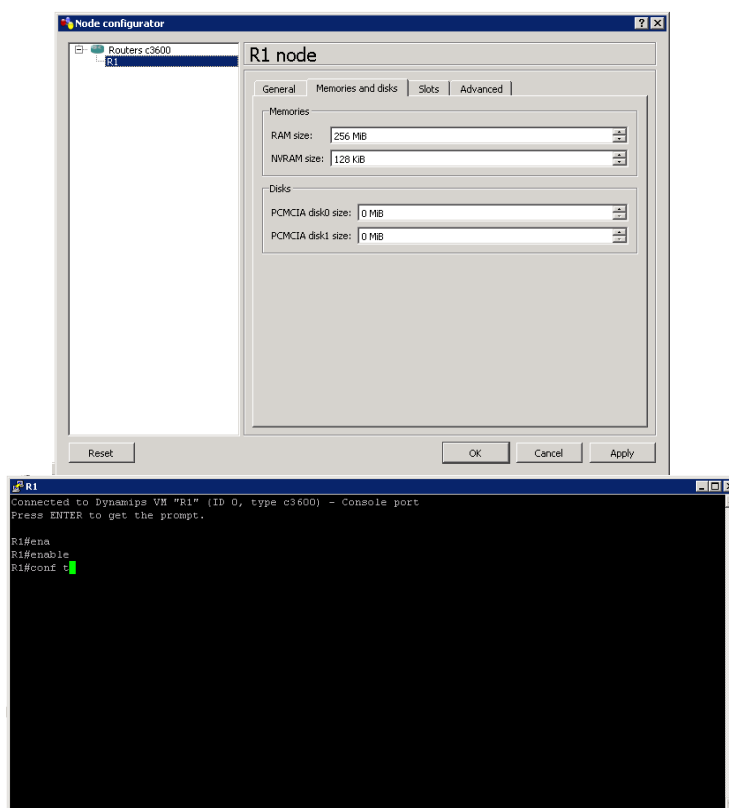


Figura 39 - Janelas de configuração de um *router*

Para além de todas estas funcionalidades, importa realçar o facto de o GNS3 permitir que um cenário virtual construído pelo utilizador possa ser interligado a uma rede real.

4.3 Conclusão

Hoje em dia existem no mercado várias ferramentas que são utilizadas para desenvolver soluções de *networking* num ambiente virtual.

Da informação disponível conclui-se que existe uma grande similaridade entre elas, mais concretamente no modo de construção das topologias, interface gráfica para configuração de dispositivos, entre outras. No entanto, umas possuem um maior número de dispositivos que podem ser utilizados para a construção do cenário, tipos de conexão, melhor usabilidade, maior credibilidade e robustez, etc., como é o caso da *Cisco Packet Tracer* e da ferramenta GNS 3.

A escolha da ferramenta mais adequada depende de vários aspetos que variam de caso para caso. No caso do trabalho apresentado nesta tese, pretendia-se um simulador que proporcionasse o maior número de funcionalidades, dispositivos, conexões, protocolos, etc., de forma a ser possível construir um cenário de rede o mais aproximado possível da rede real. Era igualmente importante escolher uma ferramenta didática, simples e credível no mercado da gestão de redes.

Quem detém algum conhecimento na área de *networking* certamente sabe que quando se pensa numa ferramenta existente no mercado para simular cenários de rede, surgem automaticamente os nomes do *Cisco Packet Tracer* e GNS 3. Tratam-se de ferramentas que são as mais utilizadas no ramo de *networking*, não só pela sua qualidade, mas também pela confiabilidade. A ferramenta a escolher para ser utilizada neste trabalho teria que ter em conta um requisito essencial para que o objetivos deste trabalho fossem alcançados: o simulador teria que suportar a utilização de máquinas virtuais. Este requisito é fundamental uma vez que é a única forma de representar na simulação a aplicação de gestão *Mobile Agent Network Management Application*. O facto do GNS 3 ser a única ferramenta a suportar tal funcionalidade, e tratando-se de uma ferramenta que permite também a comunicação com uma rede de dados real, conduziu à sua escolha.

5 Mobile Agent Network Management Application

5.1 Introdução

Neste capítulo é apresentada a arquitetura do sistema de gestão baseado em agentes móveis que é utilizada neste trabalho. Para além de serem apresentados os principais elementos que fazem parte deste sistema e a forma como estes interagem, são também referidas quais as principais limitações que este sistema apresenta e a forma como estas podem ser ultrapassadas.

5.2 Arquitetura

5.2.1 Visão geral

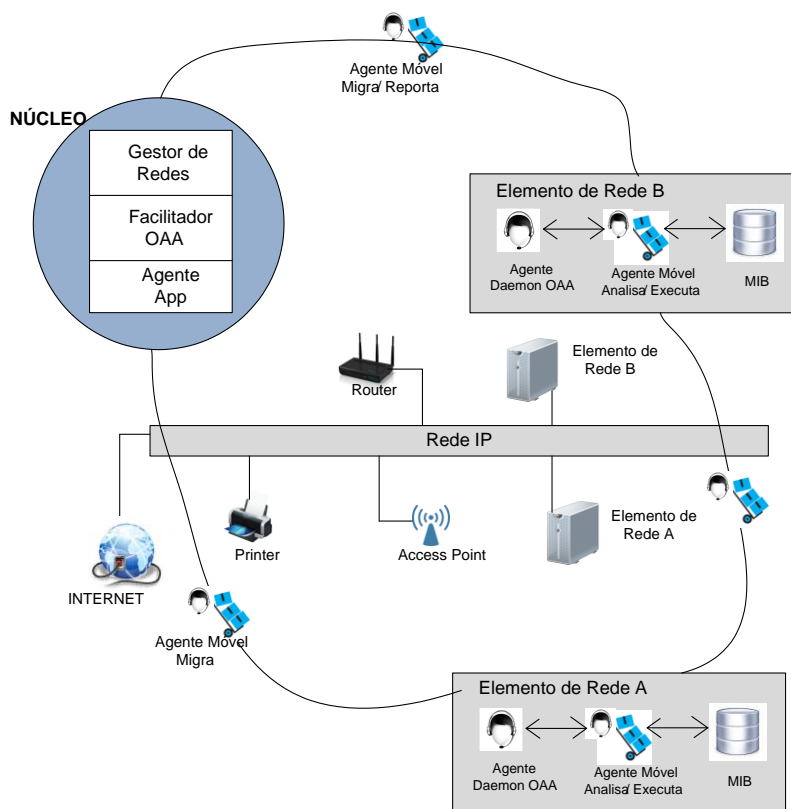


Figura 40 - Arquitetura do sistema (baseada em [32, 61])

A arquitetura deste sistema contém três elementos fundamentais, que ao analisar a figura 40, podem ser facilmente identificados. Estes elementos são: Núcleo, Elementos de Rede e os Agentes Móveis. Todos estes elementos estão inseridos numa rede IP que suporta todo o sistema [32, 61].

O Núcleo é o responsável pela gestão dos elementos de rede. Para tal, foi dotado com funções de gestão. É neste elemento que está centralizado o controlo do sistema e tem a responsabilidade não só de criar e instanciar os agentes móveis, mas também de apresentar os seus resultados. Grande parte dos elementos da rede, assim como os agentes móveis, possuem capacidades SNMP. A unificação dos serviços e capacidades de todos os elementos que fazem parte do sistema também são responsabilidade do Núcleo.

O agente Facilitador do OAA é quem detém a informação sobre todos os outros agentes existentes. Sempre que um agente é inserido na comunidade necessita de informar quais são as suas capacidades/funcionalidades ao agente Facilitador. Assim este possui conhecimento sobre todos os agentes disponíveis na comunidade assim como das funcionalidades que cada agente da comunidade detém. Quando surge um pedido para alguma tarefa, esse pedido é

feito ao agente Facilitador que direciona o pedido para o agente que possui capacidade para executar a tarefa.

O Elemento de Rede pode ser um *switch*, um servidor ou uma *workstation*. Portanto, são todos os dispositivos que estão inseridos na rede e que necessitam de gestão.

Relativamente ao agente *AgentApp* tem a capacidade de controlar, criar e monitorizar os agentes móveis. Quando este componente é inicializado pelo sistema, cria e lança um agente móvel de acordo com as preferências do utilizador. Após o seu lançamento, o estado e as ações do agente têm de ser monitorizados. De salientar que não devemos confundir estas ações de controlo com as funções do agente Facilitador, que é responsável por saber quem tem capacidade para executar o quê. É o *AgentApp* que detém a responsabilidade de receber e apresentar ao gestor de redes os relatórios produzidos pelos agentes móveis contendo a informação que estes produziram, face à(s) tarefa(s) que lhes foram atribuídas.

Os agentes móveis, estão inseridos numa comunidade de agentes denominada OAA, e detêm capacidades SNMP para que seja possível gerirem os elementos inseridos na rede. Se observarmos os elementos A e B, na figura anterior, verificamos que estes agentes analisam informação e executam funções de gestão através de interações com as respetivas MIBs. Para que isso seja possível, estes necessitam de ter capacidade de comunicar através do protocolo SNMP. Possuem descrição comportamental, informação sobre o seu estado, atributos próprios e foram criados de forma a moverem-se ao longo da rede para efetuarem trabalho de gestão em determinados *hosts* inseridos na mesma. Isto implica que seja enviado o seu estado e código. Podemos identificar outras fases da vida do agente móvel, em que este executa e/ou analisa, demonstrando inteligência, uma vez que decide qual o próximo ponto para onde se vai mover e se precisa de executar alguma ação [38].

Por último, o agente *Daemon OAA*, que apesar de não ser um agente móvel, está igualmente inserido na mesma comunidade de agentes. Este agente tem como propósito a receção dos agentes móveis dispondo funcionalidades para que estes reponham o estado e atributos.

5.2.2 Agentes Móveis da Comunidade

Como já referido anteriormente, existe uma comunidade onde estão inseridos todos os agentes que fazem parte deste sistema. Inicialmente, quando começamos a utilizar este sistema no nosso trabalho esta comunidade de agentes era constituída por 7 agentes móveis cuja responsabilidade era efetuar tarefas de auxílio à gestão da rede. Estes agentes são: *AgentCPU*, *AgentRAM*, *AgentPROC*, *AgentDISK*, *AgentSwitch*, *AgentNA* e *AgentMRTG*.

AgentCPU

É o agente responsável por monitorizar a quantidade de processamento que está a ser utilizada num determinado elemento da rede (ou vários elementos). Ou seja, devolve ao

gestor de redes uma lista de consumo de processamento dos processos que se encontram em execução no elemento de rede.

AgentRAM

Este agente tem como responsabilidade efetuar a monitorização da quantidade de RAM que está a ser utilizada num, ou vários, elementos de rede. Tem a capacidade de fazer cálculos de quantidade em bytes e percentagens de utilização de memória RAM. É proactivo ao reagir e tomar decisões relativamente ao consumo de memória.

AgentPROC

Trata-se de um agente que realiza a monitorização de processos que se encontram em estado de execução num elemento de rede, verificando, com base numa lista, se o estado dos mesmos é o desejável. Caso um processo não se encontre no estado desejável o agente efetua as ações necessárias para que o processo fique nesse estado. Todo este procedimento é feito localmente, ou seja, a verificação de cada um dos processos é feita no elemento de rede.

AgentDISK

O *AgentDISK* tem como função analisar e corrigir situações relacionadas com o espaço de armazenamento no disco de um determinado elemento de rede.

AgentSwitch

Possui a capacidade de recolher informação relativa a atributos dos equipamentos ativos de rede, mais concretamente sobre as suas interfaces. Obtém informação como a descrição dessas interfaces, a sua capacidade em termos de velocidades, número total de octetos recebidos, entre outros.

AgentNA

Este agente permite obter informação sobre a conectividade num determinado elemento de rede. Para tal, o *AgentNA* disponibiliza principalmente informação sobre o tráfego de pacotes no adaptador de rede do elemento de rede, permitindo assim analisar o seu comportamento de forma a não só corrigir problemas de conexão, como também evitar que os mesmos surjam. Esta funcionalidade torna-se especialmente útil em elementos de rede em que é fundamental estarem conectáveis de uma forma estável, como são os casos de servidores controladores de domínio, servidores *web*, etc.

AgentMRTG

O *AgentMRTG* está por norma a aguardar que sejam solicitados os seus serviços, principalmente por agentes como o *AgentCPU* e *AgentRAM*. Através do *AgentMRTG* é possível representar graficamente não só informação de gestão resultante das tarefas dos agentes,

mas também informação relativa ao elemento de rede como a sua descrição, à quanto tempo se encontra ligado, o seu endereço IP e a data/hora a que se iniciaram as suas tarefas de gestão. Ou seja, o *AgentMRTG* desloca-se ao elemento de rede, cria o gráfico com o tipo de informação que acabamos de referir e retorna esse gráfico ao gestor de redes. Com a utilização destes gráficos é possível ao gestor de redes ter uma ideia mais sugestiva e didática, daquilo que foi objeto de estudo.

5.2.3 Interação entre os Agentes

Existem duas condições iniciais que têm de ser satisfeitas para que este sistema funcione da forma correta. Essas condições são: o agente Facilitador precisa de estar em execução na estação gestora e os agentes *Daemon* têm de estar em execução nos elementos de rede que se pretende que sejam geridos. Na figura que apresentamos de seguida, figura 41, é possível visualizar estas condições de funcionamento. Satisfeitas estas condições, o sistema pode ser lançado e os agentes registados. Esses agentes indicam ao Facilitador o seu nome e quais as suas funcionalidades (*solvables* que conseguem resolver), ficando automaticamente disponíveis para executarem as tarefas que o utilizador entender.

Uma vez concluídos os passos acima referidos é possível aceder à interface da aplicação de gestão. Aqui o utilizador poderá efetuar a criação do(s) agente(s) que achar necessário(s), de forma a ver realizadas as tarefas de gestão que pretende que sejam efetuadas. Para tal, o utilizador seleciona o tipo de agente que pretende criar (ou vários agentes de diferentes tipos), quais os segmentos alvo e eventualmente se vão ser feitos testes, após a criação dos agentes móveis. Na figura 41 é possível visualizar este processo.

Todo este processo termina com o retorno dos resultados à estação gestora, sendo estes apresentados ao utilizador através de uma interface gráfica. Após estes serem apresentados, o sistema continua em execução.

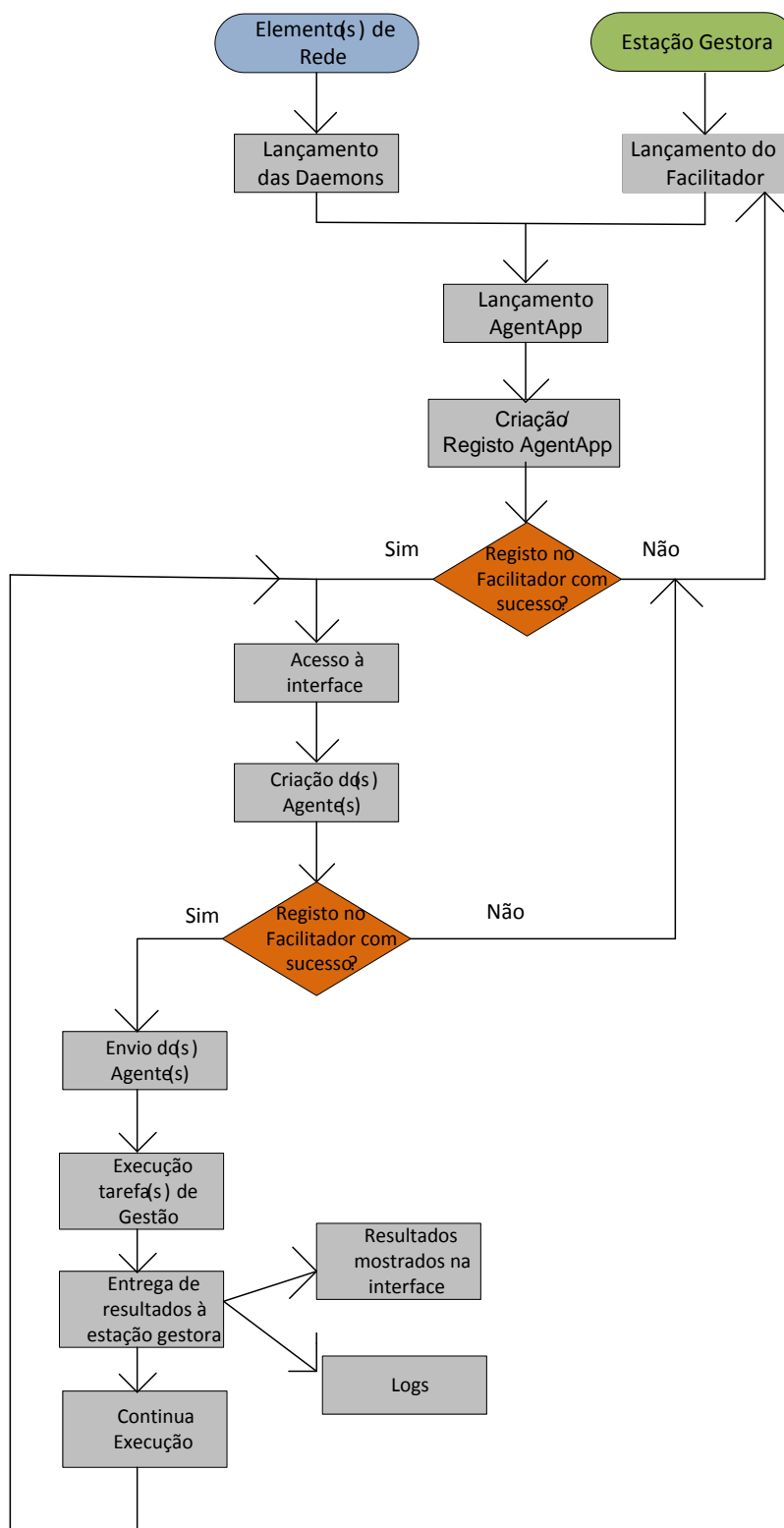


Figura 41 - Funcionamento do Sistema

Um outro aspeto importante consiste no funcionamento dos agentes móveis, principalmente no que diz respeito ao seu ciclo de vida e alterações que se vão verificando nos mesmos, ao longo do seu trajeto. A figura seguinte, figura 42 ilustra este processo.

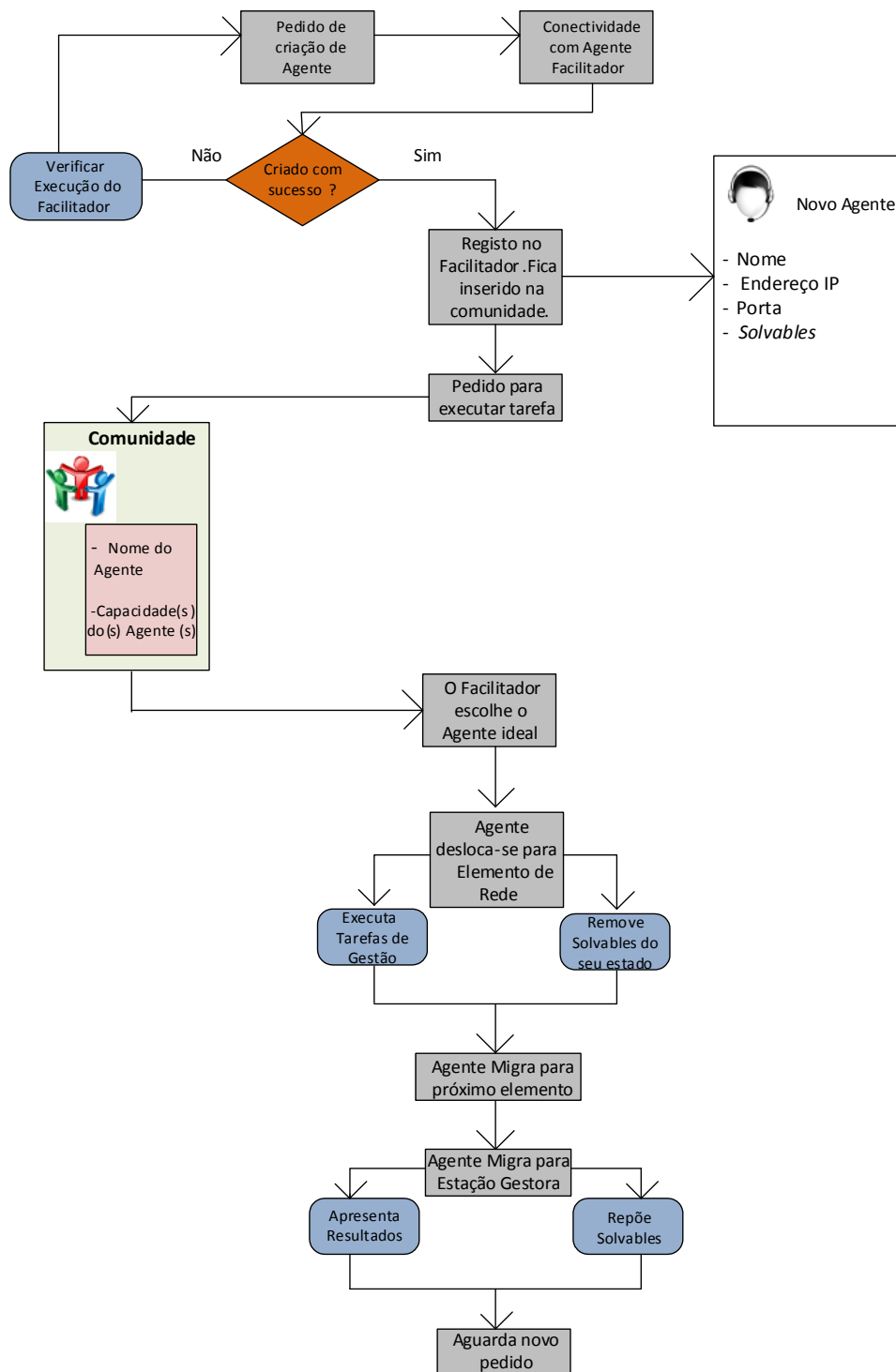


Figura 42 - Funcionamento dos Agentes Móveis

Como podemos verificar através da figura anterior é feito um pedido de criação de um agente através da interface da aplicação de gestão, sendo necessário verificar se existe ou não conectividade com o agente *Facilitador*. Caso esta não exista, é necessário verificar se o agente *Facilitador* está em execução e tentar novamente. Se existir conectividade com o

Facilitador o agente criado regista-se, indicando o seu nome, a sua localização sob a forma de endereço IP, porta de conexão e os *solvable*s que consegue resolver.

Uma vez registado na comunidade de agentes, o novo agente fica a aguardar por um pedido no qual seja necessária a sua intervenção. Quando é feito um pedido junto da comunidade de agentes, o Facilitador analisa as capacidades de cada agente e seleciona aquele mais adequado para realizar as tarefas solicitadas. Uma vez escolhido o agente, este desloca-se ao elemento de rede alvo de gestão e executa duas funções em simultâneo: as tarefas de gestão para as quais foi enviado e a remoção dos seus *solvable*s do seu estado, de forma a indicar que se encontra indisponível/ocupado para executar um novo pedido que possa eventualmente surgir enquanto se encontra a desempenhar as tarefas que lhe foram atribuídas.

No caso de o agente ter ainda de se deslocar para um outro elemento de rede, de acordo com o seu itinerário, migra para esse segundo elemento e executa as tarefas de gestão que estão definidas para esse elemento. O destino final é a estação gestora, onde apresenta os resultados da execução das tarefas que lhe foram atribuídas para que sejam visualizados na interface pelo utilizador. De seguida, volta a colocar os *solvable*s que tinha eliminado anteriormente de forma a estar novamente disponível para um futuro pedido junto da comunidade.

Existem outras noções importantes que é necessário referir, nomeadamente a forma como os agentes se movem, como interagem com as Daemons que se encontram em execução nos elementos de rede e como leem ou escrevem valores nas MIBs. Para tal, é fundamental que os agentes tenham a capacidade de comunicar via SNMP.

Para a leitura e escrita de valores na MIBs é utilizada uma API específica para o efeito. Este processo será apresentado e explicado no capítulo 6.

Na figura que apresentamos de seguida, figura 43, estão representados a migração de um agente, a interação com a Daemon que aguarda a sua chegada no elemento de rede e uma classe auxiliar que armazena o estado do agente antes de este se deslocar. O ponto de partida consiste num pedido junto da comunidade para resolver um determinado *solvable*, sendo posteriormente selecionado qual o melhor agente para satisfazer a esse pedido. Associado a esse agente está um itinerário, que consiste na sua deslocação para um elemento de rede, execução das suas tarefas e regresso à estação de gestão para apresentar os seus resultados. A migração do agente começa após ser selecionado pelo agente Facilitador.

Para se mover, um agente serve-se de uma classe auxiliar para guardar os atributos e o seu estado. Para cada tipo de agente existe uma classe desse mesmo tipo. Por exemplo, um *AgentCPU* irá procurar apoio numa classe *CPU*. É criado e instanciado um objeto desta classe, seguindo-se uma verificação de conectividade com a Daemon do elemento de rede para o qual o agente pretende deslocar-se. Se não existir qualquer problema de conectividade, é feita uma serialização dos atributos dos agentes, assim como do seu estado. No caso de haver

problemas de conectividade com a Daemon, o agente informa a aplicação de gestão sobre o problema e move-se para o elemento seguinte, conforme o seu itinerário. As Daemons, estão sempre em execução e a aguardar pedidos de conexão.

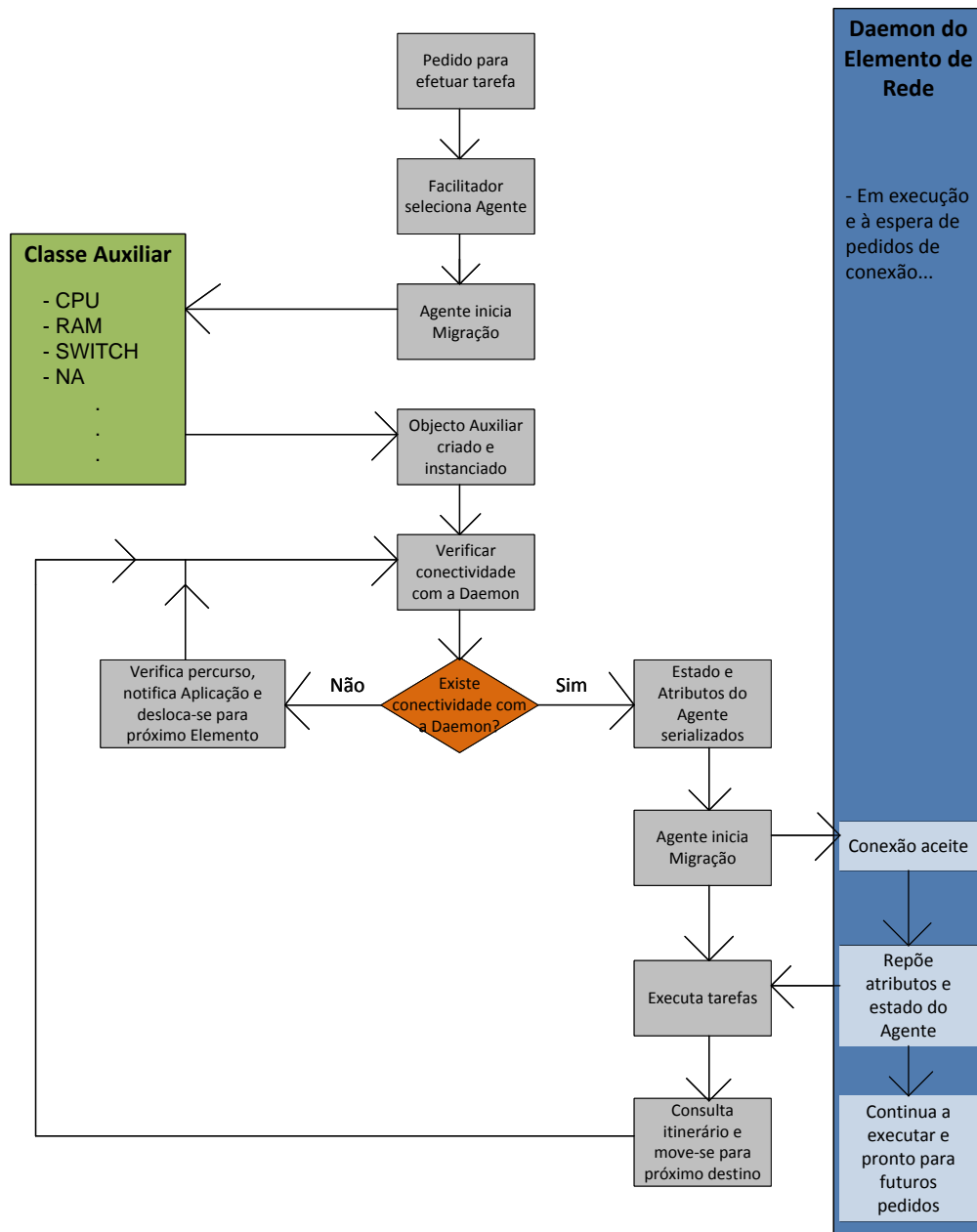


Figura 43 - Funcionamento da Migração dos Agentes no Sistema

Após este procedimento o agente dá início ao processo de migração e é feita a conexão com a Daemon, os dados são transmitidos, a Daemon repõe o estado e os atributos do agente, continua a sua execução e fica à espera de novos pedidos de conexão. O agente efetua localmente as suas tarefas e quando termina verifica o seu itinerário, podendo deslocar-se para um próximo elemento de rede ou então para a estação de gestão, conforme o próximo destino do seu percurso.

5.2.4 Limitações da Aplicação baseada em Agentes

A aplicação *Mobile Agent Network Management Application* trata-se de uma aplicação com um crescimento modular e que ainda não está na sua versão final [32, 61]. Assim sendo, apresenta algumas limitações e carências de funcionalidades que podem ser melhoradas e ultrapassadas, no sentido de ir ao encontro das necessidades e exigências da infraestrutura do GECAD.

A aplicação contém implementados agentes móveis que se encontram num estado considerado satisfatório, nomeadamente os agentes *AgentRAM*, *AgentDISK* e *AgentMRTG*. No entanto, o mesmo não acontece com os agentes *AgentCPU*, *AgentPROC*, *AgentSwitch* e *AgentNA*, sendo estes dois últimos, os que se encontram mais limitados. O agente *AgentSwitch* limita-se somente a recolher informação referente a um ativo de rede e a mostrá-la na consola. Não contém uma interface bem construída e atrativa para exibir a informação ao utilizador. Para além disso, não existe nenhuma funcionalidade implementada que permita ao utilizador armazenar essa informação de forma a estar disponível para ser consultada posteriormente, caso exista a necessidade de efetuar estudos sobre o elemento em causa. O agente *AgentNA* possui a mesma limitação. O agente recolhe a informação das MIBs, no entanto, não é utilizada para qualquer ação, impossibilitando, por exemplo, que possa ser feito por parte do gestor de redes um estudo mais eficiente e claro sobre a conectividade do adaptador de rede de um determinado elemento de rede.

Verificou-se também que existia a necessidade de aumentar a comunidade de agentes de forma a esta possuir pelo menos um novo agente que permitisse executar tarefas relacionadas com novos tipos de elementos de rede, como impressoras, telefones VoIP, etc., permitindo que a aplicação de gestão não seja limitada a ativos de rede, PCs e servidores.

A aplicação contém um método que é utilizado por todos os agentes para efetuar a leitura dos valores que se encontram nas MIBs. No entanto, através da análise dos *outputs* dos valores obtidos pelos agentes *AgentSwitch* e *AgentNA* foi possível verificar que este método não está totalmente preparado para lidar com valores do tipo *octectstring*, uma vez que os valores obtidos, desse tipo de dados, não são humanamente interpretáveis, nem surgem tal e qual como estão nas MIBs. Portanto, o método usado atualmente não é eficiente.

Outro aspeto que poderá ser melhorado refere-se à forma como o agente Facilitador é lançado. Antes de iniciar a aplicação baseada em agentes o utilizador precisa de executar manualmente o agente Facilitador. Apesar de ser um procedimento simples, consideramos que não é prático. Por outro lado, existe sempre a possibilidade de o utilizador se esquecer de iniciar o agente Facilitador antes de iniciar a aplicação, o que impedirá que esta funcione. Portanto, este procedimento pode ser simplificado de forma a melhorar a usabilidade da aplicação e também a evitar este tipo de problemas.

5.2.5 Proposta de melhoramento

Neste trabalho propõem-se as seguintes alterações à aplicação *Mobile Agent Network Management Application*: melhorar os agentes *AgentSwitch* e *AgentNA*; criar um novo agente chamado *AgentPrinter*; implementar um novo método para efetuar a leitura dos dados nas MIBs, que será utilizado pelos agentes *AgentSwitch*, *AgentNA* e *AgentPrinter*; e colocar o agente Facilitador a iniciar automaticamente quando o utilizador inicia a aplicação.

Relativamente aos agentes móveis que indicamos acima, os melhoramentos consistem em:

- *AgentSwitch* - Criar uma base de dados onde é armazenada, em forma de relatório, toda a informação que é retornada pelo agente, referente a um determinado ativo de rede, mais especificamente um *switch*. Como complemento a esta base de dados é também desenvolvida uma nova interface para apresentar o relatório ao gestor de rede, que após ser analisado, pode ser guardado na base de dados. Esta interface também permitirá ao utilizador selecionar relatórios anteriores e eliminá-los no caso de concluir que já não são necessários. Desta forma, é possível ao gestor de rede efetuar estudos sobre as tendências de fluxo na rede, entre outros, auxiliando-o na gestão e tomada de decisão.
- *AgentNA* - Criar uma interface onde será apresentada a informação dos atributos de um adaptador de rede, sendo uma parte dessa informação representada através de um gráfico. Assim, será possível ao utilizador analisar essa informação de uma forma mais atrativa, interpretar melhor os resultados e tirar conclusões sobre a utilização do elemento de rede, bem como da sua conectividade.

AgentPrinter – Desenvolver e implementar um agente capaz de efetuar a instalação automática de uma impressora num determinado computador. Assim, conseguimos alargar a comunidade de agentes, variar o tipo de capacidades/funcionalidades que estes podem desempenhar, permitindo efetuar tarefas relacionadas com recursos diferentes de computadores, servidores e ativos de rede. Considera-se como uma grande vantagem, o facto de permitir deslocar o processamento local do servidor de impressoras, para o elemento de rede, e automatizar o processo de instalação da impressora nos elementos de rede.

O novo método para a leitura das MIBs será utilizado nos agentes *AgentSwitch*, *AgentNA* e *AgentPrinter* permitindo resolver o problema que existe na obtenção correta dos valores das MIBs, mais concretamente os que são do tipo *octecstring*. Para desenvolver o novo método foi utilizada uma nova API SNMP que contém classes específicas para fazer a leitura de dados das MIBs de uma forma mais eficiente e simples, comparativamente ao método inicialmente implementado. Os valores das MIBs são apresentados na aplicação tal e qual como são lidos, independentemente do seu tipo de dados. Desta forma, para além de ficar ultrapassado o problema da interpretação de valores que referimos na secção anterior, ficamos com um método mais simples e eficiente, como será demonstrado mais à frente neste documento.

Mobile Agent Network Management Application

Relativamente ao agente Facilitador este consiste num ficheiro executável. Como tal, decidimos modificar a classe responsável por iniciar a aplicação, para que o ficheiro que representa o agente Facilitador seja executado automaticamente ao mesmo tempo que "arranca" a interface principal da aplicação.

6 Solução Proposta

6.1 Introdução

Neste capítulo são apresentados alguns detalhes de implementação das melhorias realizadas na aplicação de gestão *Mobile Agent Network Management Application*, bem como os resultados das mesmas. É também apresentado um cenário construído no simulador onde é feita a representação virtual da rede do GECAD e da nova versão da aplicação de gestão baseada em agentes móveis.

6.2 Detalhes de Implementação

6.2.1 Agente *AgentPrinter*

Como referimos anteriormente o agente *AgentPrinter* é um agente novo e que permite automatizar o processo de instalação de uma impressora num elemento de rede. Para tal, foi necessário desenvolvê-lo e efetuar a sua integração na comunidade e no sistema, construir uma interface para o agente, e implementar os *solvable*s do mesmo.

Tal como acontece com todos os agentes da comunidade o *AgentPrinter* é uma subclasse de uma classe chamada *Agent*. Esta classe *Agent* tem as seguintes variáveis de instância:

- *Myoaa* - É a variável que fica com a ligação à plataforma OAA. Esta plataforma é do tipo *Liboaa*;
- *LibcomString* - Apoia a criação da ligação à plataforma OAA;
- *AgentName* - Contém o nome do agente;

Solução Proposta

- *AgentSolvables* - Consiste nas capacidades do agente, ou seja, nas suas funcionalidades.

Para além destas variáveis de instância, a classe *Agent* contém ainda os métodos *ConnectFac* e *OaaDoEventCallback*. O primeiro é o método que irá efetuar o registo de um agente na comunidade de agentes. O segundo é o método que é chamado quando é necessário executar um *solvable* de um determinado agente.

Uma vez que a classe *AgentPrinter* é uma subclasse da classe *Agent* as variáveis e os métodos são herdados, tal como acontece com todos os agentes.

Outro agente que desempenha um papel bastante importante no sistema é o agente *AgentApp*. Trata-se do agente interface e é quem faz a "ponte" entre a comunidade de agentes e a componente de interface, permitindo comunicação em ambos os sentidos. Quando o utilizador seleciona na interface um determinado agente para efetuar uma tarefa é o agente *AgentApp* quem faz chegar esse pedido à comunidade. Nesse pedido é indicado o *solvable* que foi selecionado pelo utilizador, para que de seguida o agente Facilitador possa selecionar um agente que suporte esse *solvable*.

Para efetuar a criação do agente foi necessário criar uma classe *AgentPrinter*. Como acontece com os restantes agentes, o *AgentPrinter* guarda informações gerais relativas ao elemento de rede em que executa as tarefas de gestão. Para tal, tem como variáveis de instância:

- *ArrayList* com objetos do tipo *Printer* onde é guardada informação relevante à medida que cada um dos elementos de rede é percorrido;
- *ArrayList* denominado *MIB* onde são inseridos os OIDs dos quais se pretende obter valores para serem analisados e apresentados ao gestor de redes.

No caso deste agente os atributos lidos nas MIBs são os seguintes:

- *Printer Name* - Nome da impressora;
- *Printer Model* - Modelo da impressora.

Relativamente aos *solvables* que este agente suporta, são:

- *PresentReport* - Permite apresentar na interface toda a informação que foi sendo agregada ao longo do seu trajeto;
- *CheckPrinter* - Possui dois parâmetros (*idAgApp* e *segment*). O primeiro consiste no "id" do *AgentApp* que iniciou a comunicação entre a componente de interface e a comunidade de agentes. Quanto ao segundo identifica o segmento de rede para onde o agente se deve dirigir. Após o pedido para resolver este *solvable* ser recebido, o *AgentPrinter* inicia a migração segundo um itinerário baseado no segmento de rede que lhe foi passado e executa as tarefas de gestão associadas;

- *SayHello* - Contém um parâmetro (*Agt*) que permite que o agente possa dizer o seu nome caso lhe seja solicitado.

A classe *AgentPrinter* contém ainda implementado um conjunto de métodos que contribuem para que o agente possa executar os seus *solvable*s. Esses métodos são:

- *Snmp* - Método que permite obter os valores dos atributos que estão nas MIBs;
- *CheckNextHop* - Verifica se existe a possibilidade do agente se mover para o próximo elemento de rede que consta no seu itinerário. Inicia uma análise simples, que tem como objetivo obter informação sobre a conectividade para dar o próximo salto;
- *Move* - É chamado quando pretendemos que o agente se desloque para o próximo elemento de rede usando serialização e comunicação com *sockets*;
- *CompileReport* - Consiste num relatório final acerca da atividade do agente. Este relatório é feito baseando-se nas operações efetuadas, isto é, o que foi consultado, em que elementos de rede e quais as ações tomadas;
- *WriteBatFile* - Método que permite ao agente criar no elemento alvo de gestão o ficheiro ".bat" onde se encontram os comandos que serão executados localmente para efetuar a instalação da impressora pretendida;
- *RunInstallationCommand* - Método que permite a instalação da impressora num elemento de rede, executando o ficheiro criado através do método anterior.

Os últimos dois métodos são os mais importantes no que se refere à realização da tarefa para a qual este agente foi criado. No caso do *WriteBatFile* foram utilizadas duas classes nativas do NetBeans IDE 8.0 denominadas *File* e *Writer*. A primeira é a que nos permite criar localmente o ficheiro *PrinterInstallCommands.bat*, onde se encontram os comandos a serem executados para realizar a instalação da impressora no elemento de rede. Quanto à classe *Writer* contém um método chamado *write*, que permite que os comandos de instalação sejam inscritos no ficheiro acima referido. Relativamente ao método *RunInstallationCommand* contém uma classe chamada *ProcessBuilder* que permite criar um processo onde é passado por parâmetro o ficheiro *PrinterInstallCommands.bat* criado através dos métodos referidos anteriormente.

O sistema contém outro *package* onde estão inseridas as classes responsáveis pela interação com o utilizador, isto é, as interfaces gráficas. Neste *package* foi igualmente criada uma nova classe *PrinterResults* que consiste numa pequena interface onde são apresentados os atributos recolhidos na MIB relativos à impressora instalada, sendo também apresentada uma mensagem ao utilizador indicando se a impressora foi, ou não, instalada com sucesso no elemento de rede.

6.2.2 Novas Funcionalidades

AgentSwitch

Como já foi referido anteriormente, a aplicação contém um conjunto de pacotes de classes chamado *gecad.imanm.gui*. Uma vez que as novas funcionalidades deste agente estavam associadas à construção de uma interface específica para o mesmo, foi neste pacote que se focou o nosso trabalho. Foi criada uma nova classe chamada *Switch_Results* que consiste na interface desenvolvida e onde são apresentados os resultados do trabalho efetuado pelo agente. Grande parte desse trabalho consiste na leitura e apresentação de valores relativos a determinados atributos de cada uma das interfaces do ativo de rede, para serem analisados pelo utilizador. A informação obtida para cada uma das interfaces consiste:

- *ifDescr* - Descrição da interface;
- *ifSpeed* - Velocidade atual suportada pela interface;
- *ifOperStatus* - Permite-nos saber se a interface está, ou não, conectada a um elemento de rede;
- *ifInOctects* - Número total de octetos recebidos pela interface;
- *ifOutOctects* - Número total de pacotes transmitidos pela interface;
- *ifInDiscards* - Número total de pacotes descartados à entrada da interface, apesar de não existirem erros na sua construção;
- *ifOutDiscards* - Número total de pacotes descartados à saída da interface, mesmo não havendo erros na sua construção;
- *ifInErrors* - Número total de pacotes descartados à entrada da interface, devido a erros;
- *ifOutErrors* - Número total de pacotes descartados à saída da interface, devido a erros.

É também na classe *Switch_Results* que é feita a interação com a base de dados criada para este agente, não só para armazenar novos relatórios com os valores de gestão obtidos, como também para apresentar relatórios mais antigos que se encontram armazenados nessa base de dados. Esta classe contém como principais métodos:

- *InitializeTable* - Este método detém a responsabilidade de definir a estrutura da tabela onde os resultados são apresentados;
- *UpdateOlderStatusComboBox* - Atualiza a componente gráfica onde é apresentada a listagem dos relatórios anteriores relativos ao ativo de rede em estudo e que podem ser visualizados;

- *InitializeSwitchOlderInfo* - É executado quando a interface da classe *Switch Results* inicia. Este método é responsável por chamar o método *UpdateOlderStatusComboBox* quando a interface é executada pela primeira vez, indicando-lhe para que ativo de rede devem ser apresentados os relatórios anteriores. Assim, são apenas apresentados relatórios referentes ao ativo de rede atualmente em estudo, e não os relatórios de todos os ativos de rede.
- *FillTableForCurrentStatus* - Apresenta os valores de gestão obtidos pelo agente numa tabela para serem visualizados pelo utilizador;
- *FillTableForOlderStatus* - Apresenta, numa tabela, os valores referentes a um relatório anterior que se encontra guardado na base de dados, e que o utilizador selecionou para voltar a analisar;
- *ResetTableRows* - Este método é utilizado para limpar as linhas da tabela onde são apresentados os resultados;
- *SetParams* - É o método responsável por receber do agente os valores dos atributos do ativo de rede alvo de estudo, para que possam ser posteriormente apresentados na tabela;
- *Ready* - É chamado quando a interface inicia e invoca o método *FillTableForCurrentStatus*.

Existem ainda mais alguns métodos que estão implementados nesta classe. No entanto, não são aqui apresentados pelo facto de representarem eventos, componentes da interface, etc.

Foi, ainda adicionado ao projeto um novo pacote de classes, chamado *Database*, onde se encontram as classes que permitem efetuar todas as ações relacionadas com a base de dados criada para o agente. Essas classes são:

- *IAccessDB* - Consiste numa interface onde estão definidos os métodos que outras classes que a implementam, devem desenvolver para poderem interagir com a base de dados criada;
- *SqlServerDB* - Implementa a interface *IAccessDB*. Assim sendo, é a classe onde foram desenvolvidos os métodos que permitem inserir/eliminar/consultar os resultados do agente na base de dados;
- *FactoryAccessDB* - É responsável por instanciar um objeto da classe anterior. Este objeto é retornado, permitindo assim que a classe *Switch_Results* possa aceder à base de dados e efetuar operações na mesma.

Quanto à base de dados é possível verificar na figura seguinte, figura 44, um diagrama com as tabelas criadas, atributos e as ligações.

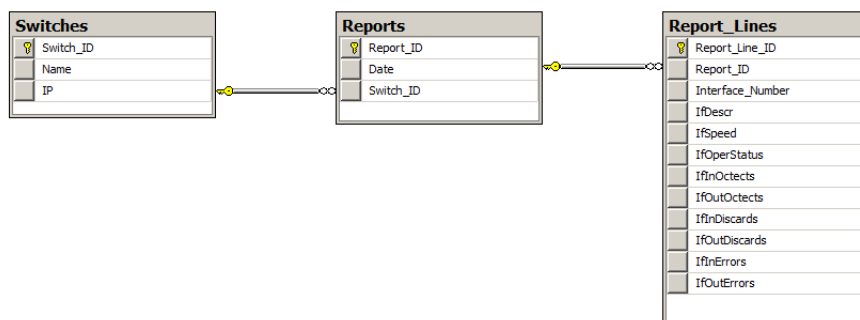


Figura 44 - Diagrama de tabelas

Analisando a figura é possível verificar três tabelas: *Switches*, *Reports* e *Report_Lines*. A tabela *Switches* contém a informação sobre o *switch* que foi alvo de gestão. Para cada *switch* é armazenado o seu nome e endereço IP. Na segunda tabela, *Reports* são armazenados os relatórios pertencentes a cada *switch* que se encontra na tabela que apresentamos anteriormente. Por último, a tabela *Report_Lines* consiste na tabela onde são armazenados os valores dos atributos analisados em cada um dos relatórios, sendo esses os valores apresentados ao utilizador quando este pretende analisar um *report*.

AgentNA

Tal como aconteceu com o agente *AgentSwitch* as novas funcionalidades deste agente foram desenvolvidas no pacote de classes *gecad.imanm.gui*. Para tal, foi-lhe adicionada uma nova classe chamada *NA_Results* que representa a interface construída para este agente, onde são apresentados os seus resultados. São obtidos os seguintes valores nas MIBs:

- *Description* - Descrição do adaptador de rede;
- *MTU* - Tamanho do maior *datagrama* que pode ser transmitido;
- *Speed* - Velocidade atual da interface;
- *MAC Adress* - Endereço físico da interface;
- *Administration Status* - Indica se a interface está *up* ou *down*, administrativamente;
- *Operation Status* - Indica se a interface está *up* ou *down*, a nível operacional;
- *Last Change* – indica a data da última alteração;
- *IfOutOctects* - Número total de octetos transmitidos pela interface;
- *IfInOctects* - Número total de octetos recebidos pela interface;

- *IfInDiscards* - Número total de pacotes descartados à entrada da interface, apesar de não existirem erros na sua construção;
- *IfOutDiscards* - Pacotes descartados à saída da interface mesmo não havendo erros;
- *IfInErrors* - Número total de pacotes descartados à entrada da interface devido a erros;
- *IfOutErrors* - Quantidade de pacotes que não foram transmitidos devido a erros.

Como já foi referido anteriormente, parte dos resultados deste agente são apresentados através de um gráfico, permitindo ao utilizador efetuar estudos de conectividade de uma forma mais simples. Este gráfico é constituído pelos últimos seis atributos apresentados acima. Para tal, foi acrescentado ao projeto da aplicação um novo pacote de classes chamado *Charts* onde estão inseridas as seguintes classes: *ChartValue* e *OutputChart*. Ambas as classes estão relacionadas, uma vez que a *ChartValue* consiste basicamente num objeto que permite modelar os dados para estarem de acordo com a estrutura esperada pela classe *OutputChart*. É esta classe que detém a responsabilidade de desenhar o gráfico na interface, utilizando a biblioteca *JFreeChart*. A classe *OutputChart* estende uma classe abstrata do *JFreeChart* denominada *ChartPanel*, permitindo assim a geração de um gráfico do tipo "Pie". Este gráfico é gerado de um modo bastante simples, uma vez que apenas é necessário fornecer os valores a representar no gráfico e as respetivas legendas.

Facilitador

O agente Facilitador consiste num ficheiro chamado "fac.exe" localizado na pasta do projeto da aplicação de agentes. Como referido na secção 5.3.4, antes de se iniciar a aplicação era necessário executar manualmente o agente Facilitador. Para ultrapassar esta limitação foi acrescentado o código à classe responsável por iniciar a aplicação, a classe *FrontEnd*. Para tal, recorreremos a duas classes nativas da *framework* de desenvolvimento, sendo estas as classes *Runtime* e *Process*.

Qualquer aplicação Java tem uma instância única da classe *Runtime* que pode ser obtida através do método *getRuntime*. Esta classe suporta um método chamado *exec* onde é passado como parâmetro o comando a executar. Para iniciar o agente Facilitador foi utilizado o comando "*cstart fac.exe*". Este método inicia um novo processo que pode ser manipulado recorrendo a um objeto do tipo *Process*, objeto esse a que demos o nome *ProcFac*. Este objeto disponibiliza métodos para a interação entre a nossa aplicação e o processo do ficheiro "fac.exe", tendo sido o *waitFor* o método utilizado para essa interação. Este método faz com que a aplicação espere que o agente Facilitador seja executado, para posteriormente arrancar e apresentar a interface principal da aplicação de agentes. Desta forma garantimos não só que o Facilitador inicia automaticamente, mas também que a aplicação de agentes não inicia sem o "fac.exe" ter sido executado, satisfazendo assim uma das premissas para o correto funcionamento do sistema.

SNMP

Como referido anteriormente, a aplicação tem implementado em todos os agentes um método que permite a obtenção dos valores que são lidos nas MIBs. No entanto, verificamos que com o agente *AgentSwitch* e o agente *AgentNA* este método não funcionava corretamente quando se tratava de valores em *octetstring*, impedindo assim que valores de determinados atributos fossem interpretados e apresentados corretamente.

Quando é pedido um determinado *solvable*, os OID referentes aos valores nas MIBs que se pretendem obter, são adicionados a um *ArrayList*, sendo posteriormente este *ArrayList* passado por parâmetro a um método denominado *snmp*, que permite obter os valores que estão nesses OIDs que recebeu. É este o procedimento em todos os agentes, sem exceção. Decidimos manter o mesmo nome para o novo método.

Relativamente ao seu desenvolvimento, foi utilizada uma API SNMP chamada "snmp4j". Para simplificar a sua utilização e de forma a programarmos num nível de abstração mais elevado, criamos uma nova classe *SNMPManager* onde foi implementado o método *obtainMibsValues*, sendo este chamado dentro do método *snmp*. O método *obtainMibsValues* permite-nos obter os valores que se encontram nas MIBs, recebendo três parâmetros:

- *IpAddr* - Consiste no endereço IP do elemento de rede ;
- *Port* - Porta do elemento de rede usada para aceder à MIB;
- *MibObjectIDs* - *ArrayList* que contém os OID que constituem a MIB, e dos quais queremos obter os valores.

Os valores obtidos pelo método são colocados num *ArrayList* de elementos do tipo *string* denominado *values*, uma vez que a API já os permite obter neste formato. Assim, é dispensado o posterior tratamento/conversão de dados, uma vez que ao estarem como *string*, já se encontram num formato perfeitamente legível e prontos a serem utilizados, para serem posteriormente apresentados na interface.

6.3 Cenário no GNS 3

Com vista a testar a solução proposta e implementada foi criado um cenário no GNS 3 onde foi virtualmente representada a nova versão da aplicação. Como se pode verificar na figura seguinte, figura 45, a representação virtual da rede do GECAD está bastante simplificada, mais concretamente no que se refere a elementos de rede, tais como servidores e computadores. A razão deve-se ao facto de se pretender utilizar máquinas virtuais para representar os *hosts* da rede. Numa rede com cerca de 50 *workstations* e 10 servidores é uma tarefa bastante morosa colocar 60 máquinas virtuais no cenário com vista a serem todos representados. No entanto, o facto de terem sido inseridas no cenário máquinas virtuais exatamente com os mesmos sistemas operativos das máquinas físicas usadas no GECAD, e respetivos serviços,

permite-nos considerar que o nosso cenário é suficiente para atingir os objetivos propostos neste trabalho.

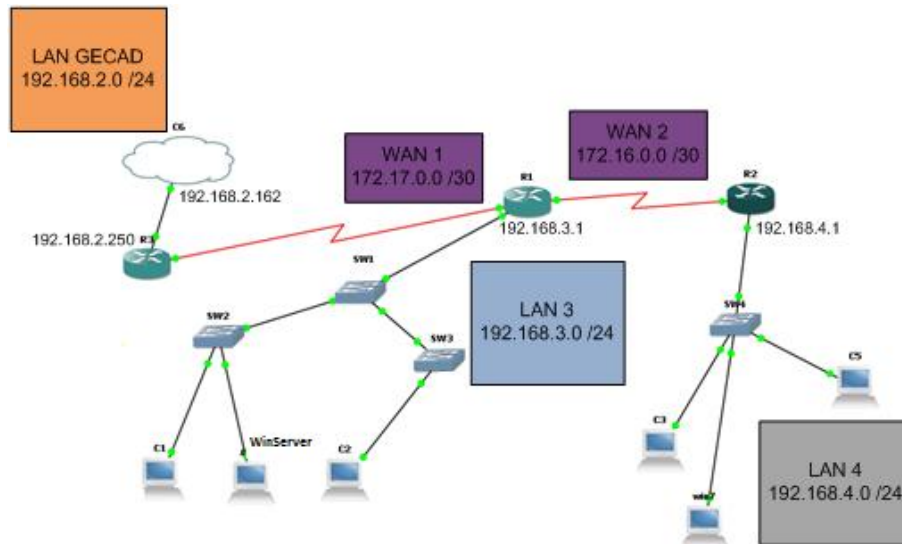


Figura 45 - Cenário criado no GNS 3

A infraestrutura do GECAD está dividida por dois edifícios dentro do ISEP, no entanto, no que à rede diz respeito, esta funciona apenas como uma única rede para ambos os edifícios. O mesmo não acontece com esta representação virtual, onde se pode verificar pela figura a existência de quatro redes distintas, tendo sido utilizada uma gama de endereçamento para cada uma delas. Esta representação foi feita com vista a ser estudada a configuração que se pretende efetuar na infraestrutura do GECAD num futuro próximo.

Nas redes virtuais foram inseridos três *routers* Cisco 3620 (R1, R2 e R3), quatro *Switches Ethernet* standard do simulador (SW1, SW2, SW3 e SW4), quatro computadores (C1, C2, C3 e C5), duas máquinas virtuais, com os diferentes sistemas operativos que são atualmente utilizados neste tipo de equipamentos, mais concretamente *Windows Server 2008* e *Windows 7*. Foi também estabelecida uma ligação à rede do GECAD através do dispositivo *cloud* (c6), representado por uma nuvem no cenário.

As máquinas virtuais foram anteriormente criadas no *VirtualBox*, tendo sido posteriormente feita a ligação com o GNS3 para que estas pudessem ser utilizadas no cenário como "*Virtual Guests*". Relativamente aos computadores, foram inseridos no cenário apenas para testar o modo de funcionamento do GNS 3 e do VPCS. Apesar de ter sido efetuado algum trabalho nestes *hosts* no sentido de os configurar para comunicarem entre si, assim como, com os restantes dispositivos inseridos no cenário, este não será apresentado neste documento, uma vez que não têm qualquer relevância nos objetivos propostos neste trabalho.

Juntamente com as máquinas virtuais, os dispositivos mais importantes deste cenário são os *routers*, uma vez que foi nestes dispositivos de rede que se focou a maior parte do nosso

trabalho na construção deste cenário. Todos os *routers* contêm o IOS C3620 versão 12.3 da Cisco. Foram ainda adicionados dois *slots* de *interfaces*, um com quatro *interfaces Ethernet* e um outro com uma interface *FastEthernet*. O tamanho de RAM e NVRAM não foram alterados, ficando os valores que o dispositivo traz por defeito, 256 MiB e 128 KiB respetivamente.

Para além das configurações mais comuns como definição de *hostname* e *passwords* de acesso, foram ainda configurados para desempenharem o papel de servidor DHCP e DNS de forma a que o processo de atribuição de endereço IP aos *hosts* inseridos numa determinada rede, seja automático. Foram ainda definidas todas as rotas necessárias para que os pacotes de dados sejam encaminhados da forma correta, permitindo assim comunicação entre todas as redes. De salientar que o facto do *router* R3 ser o grande responsável por efetuar a ligação e comunicação com a rede real, fez com que fossem necessárias mais algumas configurações para além das comuns aos restantes *routers*. Essas configurações "extra" consistiram no NAT, uma vez que o *router* do GECAD não tinha rotas para as redes virtuais que se encontram no nosso cenário, impossibilitando assim, comunicação entre redes virtuais e real. Havia duas possibilidades para resolver esta questão: ou eram adicionadas rotas ao *router* do GECAD para estas redes, permitindo saber para onde enviar as suas respostas, ou então, recorrendo ao NAT que permite traduzir os endereços dessas redes virtuais num endereço da rede 192.168.2.0. Esta última foi considerada a opção a seguir uma vez que é mais segura, principalmente no caso em questão onde o cenário virtual pode ser alterado a qualquer momento.

Como referimos anteriormente, a *cloud* (C6) é responsável por efetuar a "ponte" entre o cenário virtual e a rede do GECAD. Basicamente, representa um servidor chamado LAIDV que se encontra conectado à rede do GECAD, tendo como endereço IP o 192.168.2.162. No GNS 3, quando adicionamos a *cloud* a uma topologia, temos de indicar qual o adaptador de rede que será utilizado. Uma vez que a ferramenta foi instalada e configurada neste servidor, bastou indicar que iríamos utilizar o seu adaptador de rede real para efetuar a ligação ao exterior. Portanto, este servidor (que em termos virtuais é a *cloud* "C6") funciona como a "porta de saída" do nosso ambiente virtual e "porta de entrada" para o real.

6.4 Resultados

Como já foi referido anteriormente, o nosso cenário permite que os *hosts* que se encontram em cada uma das redes virtuais possam comunicar entre si. A figura seguinte, figura 46, apresenta um conjunto de *pings* que foram efetuados entre os vários computadores das diferentes redes, permitindo concluir que foram feitos com sucesso. Foram igualmente feitos *pings* a partir de um computador de cada rede virtual, para o *router* do GECAD (192.168.2.1), igualmente com sucesso.

```

C:\Users\administrator.GECAD\gns3\vpccs-0.16c\vpccs.exe
UPCS 1 >1
UPCS 1 >ping 192.168.4.55
192.168.4.55 icmp_seq=1 timeout
192.168.4.55 icmp_seq=2 time=42.000 ms
192.168.4.55 icmp_seq=3 time=42.000 ms
192.168.4.55 icmp_seq=4 time=42.000 ms
192.168.4.55 icmp_seq=5 time=42.000 ms

UPCS 1 >2
UPCS 2 >ping 192.168.4.33
192.168.4.33 icmp_seq=1 timeout
192.168.4.33 icmp_seq=2 time=41.000 ms
192.168.4.33 icmp_seq=3 time=41.000 ms
192.168.4.33 icmp_seq=4 time=41.000 ms
192.168.4.33 icmp_seq=5 time=41.000 ms

UPCS 1 >3
UPCS 3 >ping 192.168.3.11
192.168.3.11 icmp_seq=1 time=42.000 ms
192.168.3.11 icmp_seq=2 time=42.000 ms
192.168.3.11 icmp_seq=3 time=42.000 ms
192.168.3.11 icmp_seq=4 time=42.000 ms
192.168.3.11 icmp_seq=5 time=42.000 ms

UPCS 1 >4
UPCS 4 >ping 192.168.3.22
192.168.3.22 icmp_seq=1 time=41.000 ms
192.168.3.22 icmp_seq=2 time=39.000 ms
192.168.3.22 icmp_seq=3 time=39.000 ms
192.168.3.22 icmp_seq=4 time=39.000 ms
192.168.3.22 icmp_seq=5 time=51.000 ms

UPCS 1 >5
UPCS 5 >ping 192.168.2.1
192.168.2.1 icmp_seq=1 time=74.000 ms
192.168.2.1 icmp_seq=2 time=51.000 ms
192.168.2.1 icmp_seq=3 time=50.000 ms
192.168.2.1 icmp_seq=4 time=51.000 ms
192.168.2.1 icmp_seq=5 time=49.000 ms

UPCS 1 >6
UPCS 6 >ping 192.168.2.1
192.168.2.1 icmp_seq=1 time=54.000 ms
192.168.2.1 icmp_seq=2 time=43.000 ms
192.168.2.1 icmp_seq=3 time=42.000 ms
192.168.2.1 icmp_seq=4 time=43.000 ms
192.168.2.1 icmp_seq=5 time=31.000 ms

UPCS 1 >_

```

Figura 46 - Pings entre os computadores inseridos no cenário

Para além dos computadores, também as máquinas virtuais conseguem comunicar com outros nós virtuais e com o *router* do GECAD. Usando a máquina virtual *Win7* como exemplo, apresentamos na figura 47 essa comunicação.

```

Win7 [Em execução] - Oracle VM VirtualBox
Máquina ver Dispositivos Ajuda
C:\Windows\system32\cmd.exe
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\tese>ping 192.168.3.11

Pinging 192.168.3.11 with 32 bytes of data:
Reply from 192.168.3.11: bytes=32 time=59ms TTL=62
Reply from 192.168.3.11: bytes=32 time=43ms TTL=62
Reply from 192.168.3.11: bytes=32 time=42ms TTL=62
Reply from 192.168.3.11: bytes=32 time=40ms TTL=62

Ping statistics for 192.168.3.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 40ms, Maximum = 59ms, Average = 46ms

C:\Users\tese>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:
Reply from 192.168.2.1: bytes=32 time=91ms TTL=252
Reply from 192.168.2.1: bytes=32 time=50ms TTL=252
Reply from 192.168.2.1: bytes=32 time=63ms TTL=252
Reply from 192.168.2.1: bytes=32 time=61ms TTL=252

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 50ms, Maximum = 91ms, Average = 66ms

C:\Users\tese>_

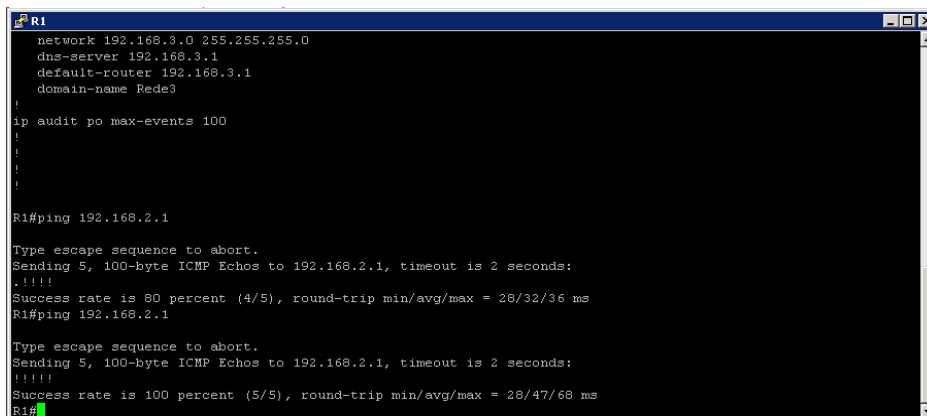
```

Figura 47 - Comunicação entre máquinas virtuais no cenário

Portanto, se os computadores e as máquinas virtuais conseguem comunicar entre eles, assim como a rede do GECAD, automaticamente todos os *routers* do cenário também o conseguem, uma vez que estes são os responsáveis pelo encaminhamento dos pacotes de dados que são

Solução Proposta

trocados pelos nós. A figura seguinte, figura 48 ilustra um dos *routers* do nosso cenário (R1) a comunicar com o router do GECAD, garantindo assim a ligação entre cenários virtual e real.



```
R1
network 192.168.3.0 255.255.255.0
dns-server 192.168.3.1
default-router 192.168.3.1
domain-name Rede3

ip audit po max-events 100

R1#ping 192.168.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/32/36 ms
R1#ping 192.168.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 26/47/68 ms
R1#
```

Figura 48 - Comunicação entre router "R1" e router do GECAD

Na figura 49 é possível verificar a interface gráfica do nosso cenário no GNS 3, onde para além de todos os dispositivos utilizados (e acima referidos), é ainda possível observar a interface do *VirtualBox* com as máquinas virtuais usadas no cenário em execução.

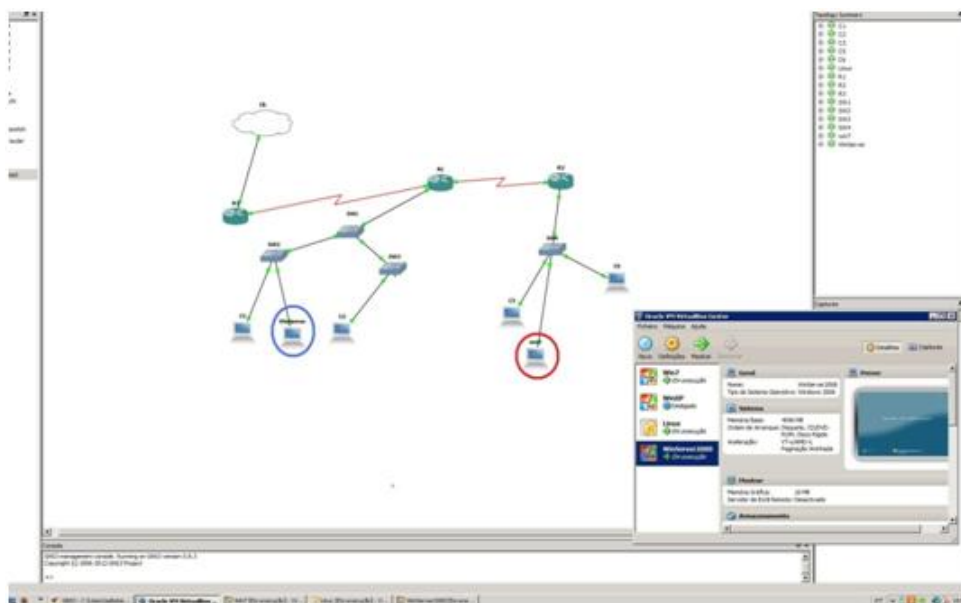


Figura 49 - Cenário no GNS 3

A nossa aplicação de agentes foi colocada na máquina virtual *WinServer* (circulo azul na figura anterior). É também neste elemento de rede que se encontra o agente Facilitador. Os agentes *Daemon* foram devidamente colocados nas máquinas virtuais, que são no fundo os nossos elementos de rede alvo de gestão.

Ao contrário do que acontecia inicialmente, quando se inicia a aplicação arranca também o "fac.exe" que representa o agente Facilitador, permitindo ser feito o registo dos agentes da comunidade. Na figura 50 é apresentada a interface principal da aplicação de agentes, bem como os registos dos agentes no agente *Facilitador*.

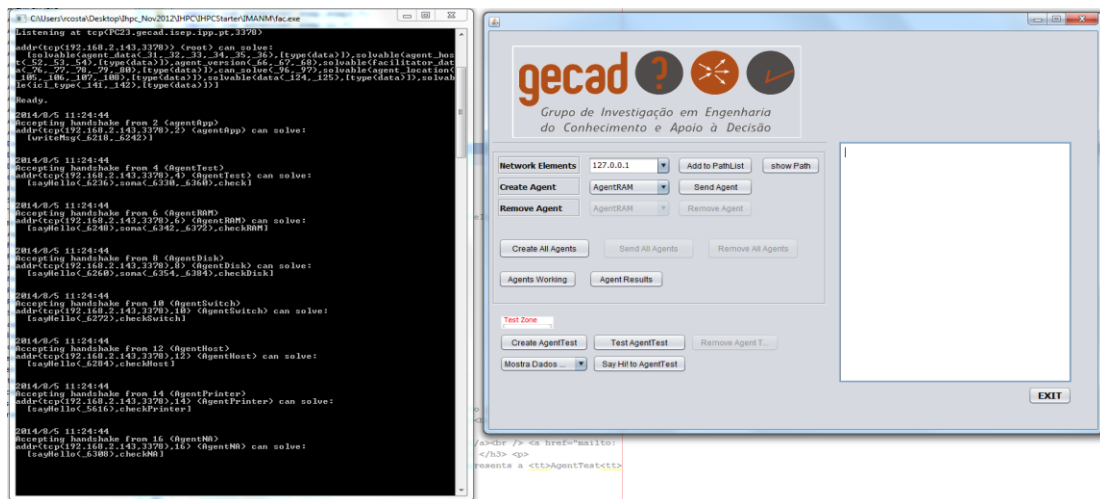


Figura 50 - Interface Principal da Aplicação de Agentes

Na figura é possível observar o registo, por parte dos agentes, dos seus *solvables* junto do agente Facilitador. Relativamente ao novo agente e aos agentes melhorados neste trabalho, os *solvables* são: *CheckSwitch*, *CheckNA* e *CheckPrinter*. É a partir desta interface que o utilizador procede à criação dos agentes para efetuar as tarefas de gestão. O procedimento para a criação dos agentes é exatamente igual para todos eles, sendo apenas necessário seguir quatro passos: selecionar na *ComboBox Create Agent* qual o agente que pretende utilizar, inserir o endereço IP do elemento de rede alvo de gestão em *Network Elements*, adicionar esse endereço ao itinerário do agente clicando em *Add to PathList* e enviá-lo, clicando no botão *Send Agent*. Este procedimento está visível na figura 51, onde é criado um *AgentNA* para efetuar um estudo de conectividade de um adaptador de rede. Esse adaptador pertence ao *host Win7* que se encontra na rede 192.168.4.0, tendo como endereço IP o 192.168.4.4.

Solução Proposta

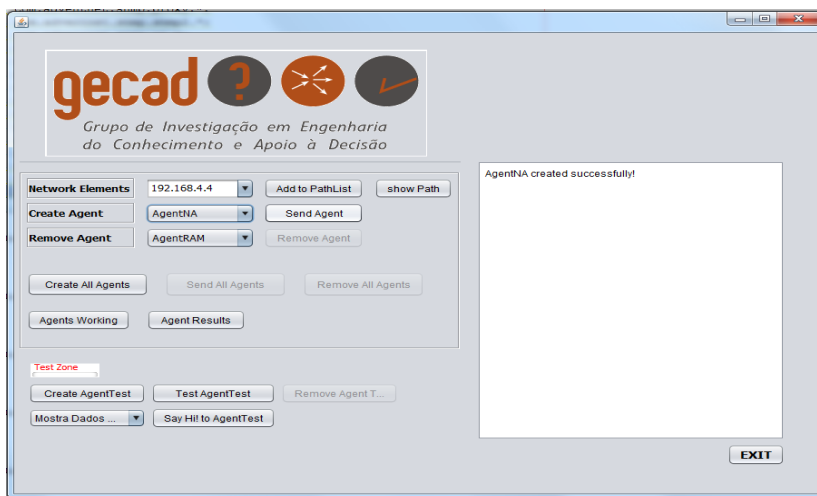


Figura 51 - Criação de um Agente na Aplicação de Agentes

Após enviar o agente este irá desempenhar as suas tarefas junto do elemento de rede. De seguida e utilizando a nova funcionalidade desenvolvida é possível ao utilizador visualizar e analisar os resultados obtidos pelo agente, sendo parte desses resultados apresentados na forma de gráfico, tal como é possível verificar na figura seguinte, figura 52.

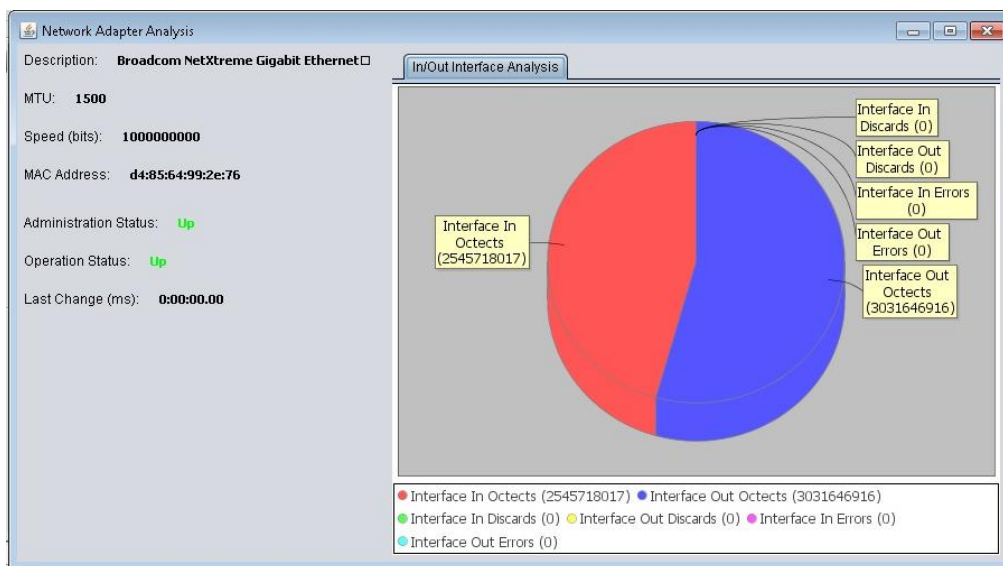


Figura 52 - Resultados do AgentNA

Como se pode concluir, a informação sobre o adaptador é apresentada de forma separada na interface. Uma parte onde é apresentado um gráfico onde o utilizador pode verificar de uma forma simples dados sobre I/O na interface de rede, e uma outra parte onde são apresentadas informações como estado administrativo e operacional, velocidade do adaptador, descrição, entre outros. No fundo, são apresentados os atributos que foram lidos das MIBs e que referimos na secção 6.2.1.

O agente *AgentPrinter* irá efetuar automaticamente a instalação de uma impressora de rede no elemento "Win7". Basicamente o agente desloca-se a esse elemento, cria e executa o ficheiro *Batch* (também conhecido por *.bat*) que contém os comandos para efetuar a instalação. Após a instalação o utilizador é informado através da interface criada para este agente e de uma mensagem, se essa instalação foi, ou não, realizada com sucesso. Nas figuras 53 e 54 apresentamos exemplos de uma instalação bem sucedida, e de uma instalação falhada, respetivamente.

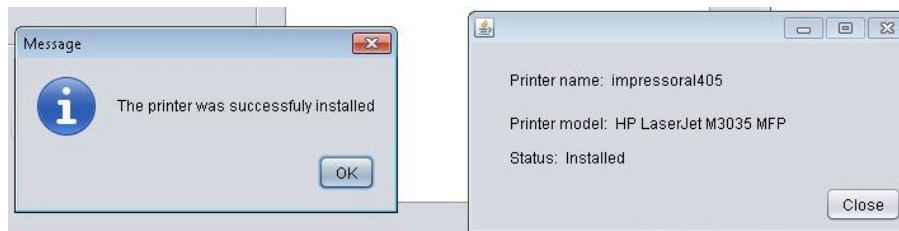


Figura 53 - Trabalho do AgentPrinter bem sucedido

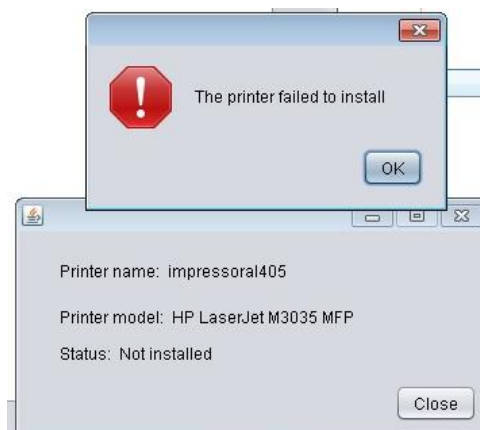
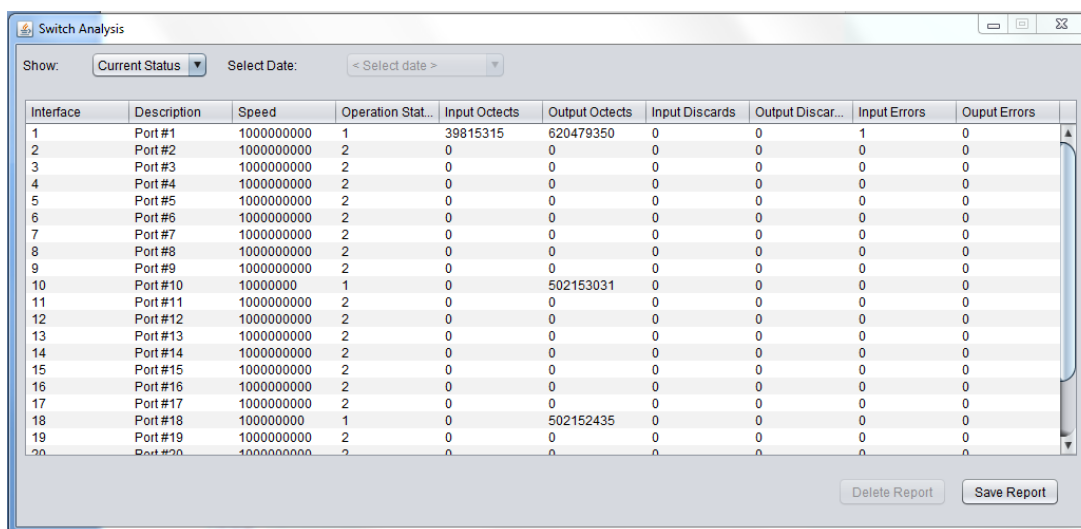


Figura 54 - Trabalho do AgentPrinter mal sucedido

No caso de o utilizador pretender analisar um *switch* o procedimento a seguir na aplicação é exatamente o mesmo que demonstramos anteriormente. No entanto, relativamente ao cenário virtual o funcionamento é diferente dos restantes agentes. Tal como indicamos na secção 4.2.3, o GNS 3 é limitado no que se refere à simulação de *switches* o que impede de criar um estudo realista de um ativo de rede deste tipo, neste cenário. O facto de o nosso cenário ter sido construído e configurado de forma a permitir comunicar com a rede do GECAD permite-nos contornar este problema. Tirando partido desta ligação é possível utilizar um *switch* que esteja inserido na infraestrutura do GECAD. Assim sendo, vamos demonstrar o funcionamento do agente *AgentSwitch* que terá como destino um HP ProCurve 1800 com endereço IP 192.168.2.21. Os seus resultados são apresentados na interface desenvolvida, como ilustrado na figura 55.

Solução Proposta

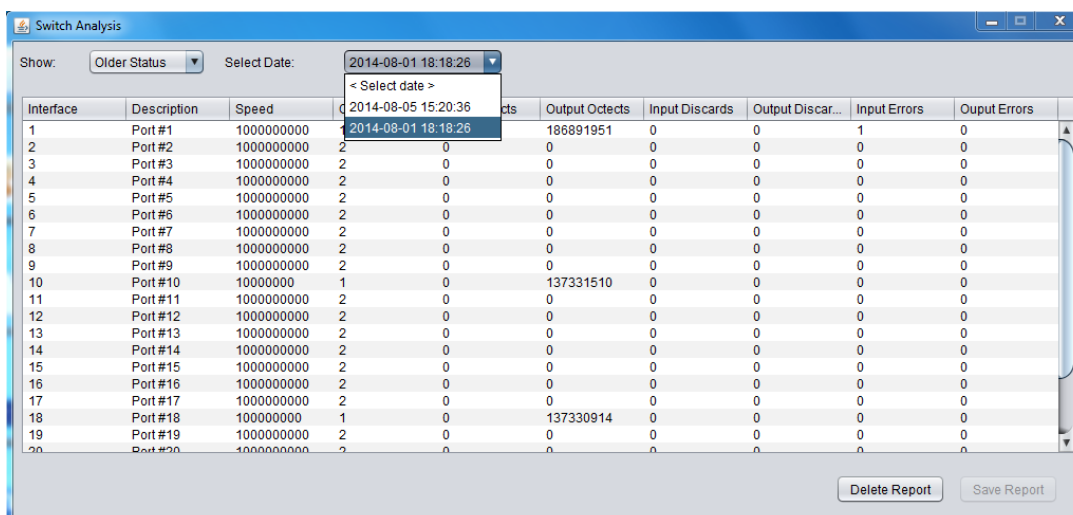


The screenshot shows the 'Switch Analysis' window with the 'Show' dropdown set to 'Current Status'. The table below displays the current status for 20 interfaces.

Interface	Description	Speed	Operation Stat...	Input Octects	Output Octects	Input Discards	Output Discar...	Input Errors	Ouput Errors
1	Port #1	1000000000	1	39815315	620479350	0	0	1	0
2	Port #2	1000000000	2	0	0	0	0	0	0
3	Port #3	1000000000	2	0	0	0	0	0	0
4	Port #4	1000000000	2	0	0	0	0	0	0
5	Port #5	1000000000	2	0	0	0	0	0	0
6	Port #6	1000000000	2	0	0	0	0	0	0
7	Port #7	1000000000	2	0	0	0	0	0	0
8	Port #8	1000000000	2	0	0	0	0	0	0
9	Port #9	1000000000	2	0	0	0	0	0	0
10	Port #10	10000000	1	0	502153031	0	0	0	0
11	Port #11	1000000000	2	0	0	0	0	0	0
12	Port #12	1000000000	2	0	0	0	0	0	0
13	Port #13	1000000000	2	0	0	0	0	0	0
14	Port #14	1000000000	2	0	0	0	0	0	0
15	Port #15	1000000000	2	0	0	0	0	0	0
16	Port #16	1000000000	2	0	0	0	0	0	0
17	Port #17	1000000000	2	0	0	0	0	0	0
18	Port #18	1000000000	1	0	502152435	0	0	0	0
19	Port #19	1000000000	2	0	0	0	0	0	0
20	Port #20	1000000000	2	0	0	0	0	0	0

Figura 55 - Resultados do AgentSwitch

Como podemos verificar, na figura anterior, os resultados retornados pelo agente são apresentados numa tabela onde o utilizador pode analisar os valores lidos nas MIBs, para cada uma das interfaces do *switch*. Aqui o utilizador poderá efetuar três ações distintas: simplesmente fechar a janela e não armazenar na base de dados o relatório apresentado; armazenar o relatório clicando no botão *Save Report*, sendo este inserido na base de dados criada para o efeito; ou então verificar relatórios anteriores. Para tal, basta ao utilizador selecionar na *ComboBox Show* (que por defeito contém a opção *Current Status*) a opção *Older Status*. Na figura 56 é possível verificar esta funcionalidade, onde o utilizador de todos os relatórios existentes na base de dados para o *switch* em causa, seleciona qual deles pretende voltar a analisar.



The screenshot shows the 'Switch Analysis' window with the 'Show' dropdown set to 'Older Status'. The 'Select Date' dropdown is open, showing a list of dates. The table below displays the statistics for the selected date.

Interface	Description	Speed	Operation Stat...	Input Octects	Output Octects	Input Discards	Output Discar...	Input Errors	Ouput Errors
1	Port #1	1000000000	2	0	186891951	0	0	1	0
2	Port #2	1000000000	2	0	0	0	0	0	0
3	Port #3	1000000000	2	0	0	0	0	0	0
4	Port #4	1000000000	2	0	0	0	0	0	0
5	Port #5	1000000000	2	0	0	0	0	0	0
6	Port #6	1000000000	2	0	0	0	0	0	0
7	Port #7	1000000000	2	0	0	0	0	0	0
8	Port #8	1000000000	2	0	0	0	0	0	0
9	Port #9	1000000000	2	0	0	0	0	0	0
10	Port #10	10000000	1	0	137331510	0	0	0	0
11	Port #11	1000000000	2	0	0	0	0	0	0
12	Port #12	1000000000	2	0	0	0	0	0	0
13	Port #13	1000000000	2	0	0	0	0	0	0
14	Port #14	1000000000	2	0	0	0	0	0	0
15	Port #15	1000000000	2	0	0	0	0	0	0
16	Port #16	1000000000	2	0	0	0	0	0	0
17	Port #17	1000000000	2	0	0	0	0	0	0
18	Port #18	1000000000	1	0	137330914	0	0	0	0
19	Port #19	1000000000	2	0	0	0	0	0	0
20	Port #20	1000000000	2	0	0	0	0	0	0

Figura 56 - Visualização de relatórios antigos do AgentSwitch

Importa ainda salientar que foram tidas em conta a usabilidade e qualidade da interface. Analisando as duas figuras anteriores é possível verificar que as opções disponíveis para o

utilizador variam em função da tarefa a desenvolver. Quando um relatório é apresentado pela primeira vez como acontece na situação ilustrada na figura 55, a única opção que está disponível para o utilizador é a *Save Report*. Quando o utilizador pretende visualizar *Older Reports*, então faz todo o sentido que passem a estar disponíveis um maior número de opções, permitindo que possam ser escolhidos os relatórios anteriores para visualizar, assim como eliminar esses mesmos relatórios. No entanto, a opção para salvar o relatório que se está a visualizar não está disponível, uma vez que não fazia sentido o utilizador poder guardar um relatório que já foi anteriormente armazenado.

7 Conclusões e Trabalho Futuro

7.1 Resumo

Este trabalho procura responder às oportunidades e desafios que vão surgindo da aplicabilidade do paradigma dos agentes móveis à gestão de redes. Foi utilizado um simulador para representar a rede do GECAD e a versão melhorada da aplicação de gestão baseada em agentes móveis. Desta forma, é possível efetuar estudos e analisar diferentes soluções de gestão em cenários virtuais.

Com vista a atingir os objetivos delineados no início deste trabalho, foi feito, na fase inicial, um breve estudo sobre o tema *Gestão de Redes*. Este estudo abrange por inteiro o segundo capítulo deste documento e foi bastante útil para uma melhor familiarização e compreensão de conceitos essenciais para uma abordagem a novos temas. Foram devidamente identificadas as áreas funcionais da gestão de redes, assim como os dois modelos relativos às arquiteturas de gestão, mais concretamente: centralizadas e descentralizadas. Os cenários ilustrativos de cada um dos modelos, que foram apresentados, permitiram identificar e entender de uma forma clara a grande diferença entre eles. Foram também estudados os modelos de gestão: informação e relacional. Com o primeiro foi possível saber em pormenor como aceder à informação de gestão presente em cada recurso, assim como à forma como esta está estruturada. Quanto ao modelo relacional descreve como é que o gestor e o sistema gerido comunicam.

De seguida foi realizado um estudo sobre a aplicação do paradigma dos agentes móveis na gestão de redes, estudo esse apresentado no Capítulo 3. Foi apresentada uma definição de agente de *software*, assim como, quais as características que este deve possuir para ser considerado um agente móvel. Foram igualmente estudados os sistemas multiagente, a sua comunicação e as arquiteturas com comunicação direta ou assistida, sendo esta última a utilizada na aplicação de agentes utilizada e melhorada neste trabalho. Por último, foi fundamental terminar o capítulo com uma introdução aos agentes móveis, tendo sido analisadas as suas características, vantagens relativamente a uma arquitetura cliente-servidor, plataformas existentes, entre outros.

Foi efetuado um estudo sobre ferramentas de simulação de redes, Capítulo 4, onde após uma pesquisa sobre o tema, e consulta de alguns especialistas neste tipo de ferramentas, foram selecionadas três possíveis alternativas a serem usadas neste trabalho. Foram analisadas as suas principais características, modo de funcionamento, de que forma se enquadravam no caso particular deste trabalho com vista a atingir os objetivos propostos, etc. Existia a necessidade de utilizar uma ferramenta que permitisse efetuar a representação virtual da aplicação de agentes, assim como, de alguns elementos de rede para gestão, como workstations e servidores com as mesmas características e funcionalidades das máquinas reais utilizadas no GECAD. A solução encontrada consiste na utilização de máquinas virtuais. O facto de o GNS 3 ser a única ferramenta que suporta tal funcionalidade, juntamente com a credibilidade e características que a ferramenta tem, conduziu a que esta fosse a ferramenta escolhida para ser utilizada neste trabalho.

Seguiu-se um estudo aprofundado da aplicação baseada em agentes móveis com vista a identificar as suas principais limitações (Capítulo 5). Foi analisada a arquitetura geral do sistema, quais os agentes móveis que se encontravam já implementados e que tarefas desempenhavam. Foram estudadas as principais classes, métodos e variáveis de instância da aplicação, assim como o seu funcionamento.

O estudo efetuado permitiu verificar as principais limitações e identificar a forma como estas poderiam ser ultrapassadas. Foram também definidas novas funcionalidades a serem adicionadas à aplicação de forma a torná-la mais eficiente, útil e completa para ser usada na gestão de redes.

Após ter sido implementado o novo agente, bem como as novas funcionalidades nos agentes já existentes, passamos a dispor de uma nova aplicação de gestão.

Com vista a validar a solução proposta foi construído um cenário de rede usando a ferramenta escolhida. Para tal, foi feita a representação virtual da rede do GECAD e da aplicação de agentes.

Por último, foram ilustrados vários exemplos de aplicabilidade quer do novo agente proposto e desenvolvido, o agente *AgentPrinter*, quer das novas funcionalidades dos agentes já existentes, agentes *AgentSwitch* e *AgentNA*.

Para concluir, foram feitos alguns testes à aplicação, no cenário criado, com o objetivo de analisar o comportamento da solução proposta.

7.2 Objetivos Alcançados

No início deste trabalho foram delineados três grandes objetivos que consistiam em melhorar o desempenho da aplicação *Mobile Agent Network Management Application*, desenvolver novos agentes móveis e propor um simulador de redes onde fosse possível representar

virtualmente a rede do GECAD e a aplicação acima referida, já com as novas funcionalidades implementadas.

No que se refere aos dois primeiros objetivos, foram identificadas as limitações existentes na aplicação e de que forma estas poderiam ser ultrapassadas. Como referido na secção 5.2.4, os agentes *AgentSwitch* e *AgentNA* estavam incompletos, não fazendo mais do que leituras às MIBs. Para além disso, não existia nenhum agente que desempenhasse tarefas em elementos de rede, diferentes de ativos de rede, workstations e servidores. Com as novas funcionalidades desenvolvidas nos agentes *AgentSwitch* e *AgentNA*, estes agentes passaram a estar completos e a poder desempenhar as suas tarefas de gestão num elemento de rede, sendo os seus resultados apresentados em interfaces personalizadas para cada um dos agentes. No caso do *AgentSwitch*, para além de poder visualizar informação de gestão sobre um determinado *switch* passou também a ter a possibilidade de armazenar essa informação numa base de dados, de forma a esta poder ser novamente analisada mais tarde. Quanto ao agente *AgentNA*, este veio facilitar a interpretação e análise de utilização e conetividade de um determinado adaptador de rede, uma vez que a informação mais importante é apresentada através de um gráfico simples e de fácil interpretação.

A criação do agente *AgentPrinter* contribuiu não só para alargar a comunidade de agentes, uma vez que não existia na solução anterior, como também para aumentar as opções disponíveis relativamente às tarefas de gestão, sendo agora possível efetuar tarefas referentes a outros tipos de elementos de rede, que não eram abrangidos inicialmente.

Relativamente ao simulador de redes, a ferramenta selecionada, GNS 3, permitiu representar a rede do GECAD e a aplicação de gestão com vista a ser possível efetuar estudos num cenário virtual o mais realista possível. Foram paralelamente criadas as máquinas virtuais no *VirtualBox* e feita a interligação com o GNS 3 para que estas pudessem ser utilizadas no cenário, e conseqüentemente colocar a nova versão da aplicação de agentes numa dessas máquinas para efetuar a sua representação virtual. Importa ainda voltar a referir que o cenário construído, já contempla modificações naturais que virão brevemente a ser efetuadas na rede física do GECAD.

Para concluir convém salientar que ficou demonstrado, não só nesta secção, mas também ao longo da secção 6.3 desta Tese, que os principais objetivos estabelecidos no início deste trabalho foram alcançados com sucesso.

7.3 Limitações e Trabalho Futuro

Este trabalho procura permitir a representação virtual da rede do GECAD, bem como de uma nova versão da aplicação de gestão *Mobile Agent Network Management Application*.

Conclusões e Trabalho Futuro

Embora, como já foi referido, este trabalho tenha contribuído para a evolução da aplicação baseada em agentes com o aumento das funcionalidades existentes nos agentes e a criação de um novo agente, esta aplicação pode ainda ser melhorada, mais concretamente sugerimos:

- Alterar a interface onde são apresentados os resultados do agente *AgentSwitch* para permitir ao utilizador selecionar o tipo de tráfego que pretende analisar, assim como, representá-los através de um gráfico como acontece no agente *AgentNA*;
- Alterar a interface referente ao agente *AgentNA* para que seja possível ao utilizador verificar os resultados relativos a um adaptador de rede, em mais que um tipo de gráfico. Poderá ser dada a possibilidade ao utilizador de escolher qual o tipo de gráfico pretendido;
- Melhorar o comportamento do agente *AgenPrinter*, para que este possa estar preparado para efetuar automaticamente a instalação de uma qualquer impressora, num determinado elemento de rede, e não uma impressora em específico.

Paralelamente, existem evoluções que podem ser feitas na aplicação como um todo, como por exemplo, existir a possibilidade de automatizar o processo de obtenção dos OIDs. Na aplicação atual, os OIDs são fornecidos manualmente, isto é, é o programador que indica quais os OIDs que serão lidos para cada elemento de rede. Apesar destes serem iguais para qualquer elemento, e desde que as MIBs usadas sejam as mesmas, quando se pretende gerir recursos como *switches* ou *routers*, o processo não é assim tão simples, uma vez que estes poderão possuir diferentes números de interfaces. Por exemplo, se o gestor de redes pretender analisar um *switch* de 24 portas e de seguida um outro *switch* de 48 portas, terá que acrescentar, manualmente no código da aplicação, os OIDs necessários para que o agente consiga obter toda a informação relativa às 48 interfaces e desempenhar as suas tarefas. Sugere-se que os OIDs estivessem disponíveis numa base de dados, sendo retornadas ao agente em função do objeto de gestão em questão.

Paralelamente, existe sempre a possibilidade de serem acrescentados novos agentes com novas funcionalidades para auxiliar o utilizador na gestão de redes. Seria interessante acrescentar dois novos agentes: um capaz de colocar automaticamente um novo elemento de rede no domínio "GECAD"; e outro de efetuar tarefas de gestão em telefones VoIP, permitindo assim, continuar a alargar as funcionalidades de gestão incorporadas na aplicação de gestão baseada no paradigma dos agentes móveis.

Referências

- [1] (2010). "Gerenciamento SNMP", Available: <http://pt.scribd.com/doc/7231009/Gerenciamento-SNMP>
- [2] A. Clemm, "Network management fundamentals". Indianapolis, Ind.: Cisco Press, 2007.
- [3] P. J. Martins, "Comparação dos Paradigmas Cliente/Servidor e Agentes Móveis: Um estudo em Gerência de Redes", MsC, Universidade Federal de Santa Catarina, 2002.
- [4] H. Luo, "Agent-based Network Management System", MsC, Department of Computer Science, Beijing Institute of Technology, Beijing, 2000.
- [5] G. Goldszmidt and Y. Yemini, "Distributed management by delegation", *Proceedings of the 15th International Conference on Distributed Computing Systems*, pp. 333-340, 1995.
- [6] F. E. B. d. M. Nogueira, "Gestão de Redes Um Serviço de Valor Acrescentado", MsC, Departamento de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, 2004.
- [7] D. Perkins, "RMON: remote monitoring of SNMP-managed LANs": Prentice Hall PTR, 1999.
- [8] T. Fioreze, *et al.*, "Comparing web services with SNMP in a management by delegation environment", *Integrated Network Management IX*, pp. 601-614, 2005.
- [9] J. Schonwalder and J. Quittek, "Secure management by delegation within the Internet Management Framework", *Integrated Network Management Vi*, pp. 687-700, 1999.
- [10] S. F. Bush and A. B. Kulkarni, "Active networks and active network management: a proactive management framework": Kluwer Academic/Plenum Publishers, 2001.
- [11] M. Bernichi and F. Mourlin, "Network Management with mobile agent toolkit", Third 2008 International Conference on Convergence and Hybrid Information Technology, Vol 2, Proceedings, pp. 1012-1017, 2008.
- [12] P. Braun and W. Rossak, "Mobile agents : basic concepts, mobility models, and the Tracy toolkit". San Francisco, CA: Elsevier : Morgan Kaufmann ; Heidelberg : Dpunkt.verlag, 2005.
- [13] D. Chess, *et al.*, "Mobile agents: Are they a good idea?", *Mobile Object Systems*, vol. 1222, pp. 25-45, 1997.
- [14] D. Gavalas, *et al.*, "A mobile agent platform for distributed network and systems management", *Journal of Systems and Software*, vol. 82, pp. 355-371, Feb 2009.
- [15] B. Liu, *et al.*, "Agent cooperation in multi-agent based network management", Proceedings of the 8th International Conference on Computer Supported Cooperative Work in Design, Vol 2, pp. 283-287, 2004.
- [16] S. Makki and S. V. Wunnava, "Application of mobile agents in managing the traffic in the network and improving the reliability and quality of service", *IMECS 2006: International Multiconference of Engineers and Computer Scientists*, pp. 32-35, 2006.
- [17] F. A. C. e. S. Mouta, "Ambiente de Desenvolvimento de Sistemas Multiagente para Controlo e Supervisão de Processos em Aplicações Distribuídas", PhD, Faculdade de Engenharia Universidade do Porto, Porto, 1996.
- [18] Y. Shi, *et al.*, "A mobile agent- and policy-based network management architecture", in *Computational Intelligence and Multimedia Applications*, 2003. ICCIMA 2003. Proceedings. Fifth International Conference on, 2003, pp. 177-181.
- [19] S. R. Stuart Wagner, Keith Landgraf, "Agent Technology for Network Management", I. Telcordia Technologies, Ed., ed, 2002.
- [20] L. Tang and B. Pagurek, "A comparative evaluation of mobile agent performance for network management", *Ninth Annual Ieee International Conference and Workshop on the Engineering of Computer-Based Systems, Proceedings*, pp. 258-267, 2002.
- [21] I. Satoh, "Building reusable mobile agents for network management", *Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, vol. 33, pp. 350-357, Aug 2003.
- [22] A. M. Souza, "Gerenciamento de Redes Baseado em Agentes Móveis", *Tecnólogo em Processamento de Dados*, Centro de Ciências Formais e Tecnologia, Universidade Tiradentes, Aracaju, Brasil, 2001.

- [23] H. Li and L. Meng, "Reconstruction Mechanism Based on Distributed Intelligent Agents for Network Management", *2008 7th World Congress on Intelligent Control and Automation, Vols 1-23*, pp. 5143-5147, 2008.
- [24] E. Y. L. Timon C. Du, An-Pin Chang, "Mobile Agents in Distributed Network Management", *Communications of the ACM*, vol. 46, 2003.
- [25] S. Aidarous and T. Plevyak, "Managing IP networks: challenges and opportunities": John Wiley, 2003.
- [26] S. B. Morris, "Network management, MIBs and MPLS". Upper Saddle River, N.J. ; London: Prentice Hall PTR, 2003.
- [27] N. Greenfield, "FCAPS Management for the Smart Grid High-level Summary", A. I. S. Engineering, Ed., ed, 2009.
- [28] M. A. Miller, "Managing internetworks with SNMP", 2nd ed. ed. New York: M&T Books, 1997.
- [29] F. S. Systems, "FCAPS White Paper", ed, 2005.
- [30] C. A. Koon-Seng Lim, Rolf Stadler, "Decentralizing Network Management", *IEEE electronic Transactions on Network and Service Management*, 2009.
- [31] M. J. Viamonte, "SMI e MIBs", ISEP, Ed., ed. Porto, 2008.
- [32] J. R. Teixeira, "Uma aplicação de Gestão de Redes Baseada no Paradigma dos Agentes Móveis", ISEP, Ed., ed. Porto, 2011.
- [33] M. J. Viamonte, "SMI e MIBs", ISEP, Ed., ed. Porto, 2008.
- [34] "MIB Browser", Available: <http://ireasoning.com/mibbrowser.shtml>
- [35] AdventNet. (2007). "ADVENTNET SNMP UTILITIES 5", Available: <http://www.webnms.com/snmputilities/help.html>
- [36] M. J. Viamonte, "SNMP – Simple Network Management Protocol", ISEP, Ed., ed. Porto, 2008.
- [37] M. J. Viamonte, "SMI e MIBs", ISEP, Ed., ed. Porto, 2008.
- [38] D. Grimshaw, "CPS 720 Artificial Intelligence Programming", Ryerson University, Toronto, Canada, 2001
- [39] C. R. António Silva, "Sistemas Baseados em Agentes", ISEP, Ed., ed. Porto, 2008.
- [40] A. S. Carlos Ramos, "Agent-Based Systems", ISEP, Ed., ed. Porto, 2008.
- [41] L. M. Helder Coelho, "Agent Workshop", F. d. C. d. U. d. Lisboa, Ed., ed. Lisboa.
- [42] L. P. Reis, "Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico", PhD, FEUP, Porto, 2003.
- [43] V. H. Barros, "Sistemas Baseados em Agentes - Agentes Móveis", ISEP, Porto, 2008.
- [44] A. R. Cardoso, "Integrando Agentes Móveis com Sistemas Legados para Gerenciamento de Redes ATM Heterogêneas", MsC, Universidade Federal de Campina Grande, Campina Grande 2002.
- [45] D. B. Lange and M. Oshima, "Programming and deploying Java mobile agents with Aglets": Addison-Wesley, 1998.
- [46] L. Zhao, "Java-Based Mobile Agents", S. o. E. a. C. Science, Ed., ed. Wellington, New Zealand.
- [47] S. Fernandes, "Introdução aos Agentes Autónomos, Agentes Móveis", INESC, Ed., ed, 2000.
- [48] M. J. Viamonte, "Novas Arquitecturas de Gestão", ISEP, Ed., ed. Porto, 2008.
- [49] S. Pleisch, "State of the Art of Mobile Agent Computing - Security, Fault Tolerance, and Transaction Support", IBM Research, Zurich Research Laboratory, Ruschlikon, Switzerland 1999.
- [50] E. Lourenço, "Agentes Móveis", U. F. d. Paraná, Ed., ed. Curitiba, 2006.
- [51] T. Sundsted. (1998). "Agents on the move", Available: <http://www.javaworld.com/javaworld/jw-07-1998/jw-07-howto.html>
- [52] A. R. Cardoso, "Integrando Agentes Móveis com Sistemas Legados para Gerenciamento de Redes ATM Heterogêneas", MsC, Universidade Federal de Campina Grande, Campina Grande 2002.
- [53] D. B. Lange and M. Oshima, "Programming and deploying Java mobile agents with Aglets": Addison-Wesley, 1998.
- [54] J. K. Chun, *et al.*, "Network management based on PC communication platform with SNMP and mobile agents", *22nd International Conference on Distributed Computing Systems Workshop, Proceedings*, pp. 222-227, 2002.
- [55] L. D. L. Gibeon Soares de Aquino, "Agentes Móveis", C. d. I. d. UFPE, Ed., ed. Recife, Brasil.
- [56] R. R. Gudwin. (2009). "Agentes Móveis", Available:

<http://www.dca.fee.unicamp.br/~gudwin/courses/IA009/Aulas.html>

- [57] J. Hu, et al., "A Network Management Model Based on Mobile Agent System", 2008 International Conference on Apperceiving Computing and Intelligence Analysis (Icacia 2008), pp. 279-283, 2008.
- [58] X. Huang, et al., "Study of the Active Network Management System Model based on Agent", 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, Vols 1-31, pp. 11165-11168, 2008.
- [59] S. K. Huy Hoang To, Bala Srinivasan, "Mobile Agents for Network Management: When and When Not!", presented at the 2005 ACM Symposium on Applied Computing, 2005.
- [60] C. P. Iwan Adhicandra, Ebrahim Shaghoei, "Using Mobile Agents to Improve Performance of Network Management Operations", presented at the PostGraduate Networking Conference, Liverpool John Moores University, 2003.
- [61] J. R. Teixeira, M. J. Viamonte, "IMANetMS - An Intelligent Mobile Agent-Based Network Management Approach for a Real Scenario", Journal of Advances in Computer Networks (JACN), Vol.1 (2), pp. 99-104, 2013.