Universidade do Porto

Faculdade de Engenharia

FEUP

Serviço de Informação LDAP no suporte à Gestão de Redes

Jorge Manuel Coutinho Soares

Porto, Dezembro 2005

FACULDADE DE ENGENHARIA DA
UNIVERSIDADE DO PORTO

Serviço de Informação LDAP no suporte à Gestão de Redes

Jorge Manuel Coutinho Soares

(Licenciado em Engenharia Sistemas e Informática

pela Universidade do Minho)

Dissertação submetida para satisfação

parcial dos requisitos do grau de

Mestre em Redes e Sistemas de Comunicação

Realizada sob a orientação do

Professor Doutor Raúl Filipe Teixeira de Oliveira,

do Departamento de Engenharia Electrotécnica e de Computadores

da Faculdade de Engenharia da Universidade do Porto

Porto, Dezembro 2005

Dedico este trabalho a duas pessoas
muito importantes na minha vida.
À minha ***mãe*** *pela coragem que me deu,*
e pela vontade de ver no seu filho este grau
académico, e à minha ***esposa*** *pelo apoio,*
carinho, paciência e sacrifícios por mim
efectuados.

Resumo

À medida que o tamanho da rede continua a crescer, multiplicam-se os equipamentos de rede, que necessitam de ser eficientemente geridos, pelo que mais informação de rede deve ser gerida, a qual suscita uma situação de gestão ineficaz. Outra questão muito importante é que tecnologias de rede heterógeneas devem co-existir e inter-operar, de maneira que as infraestruturas de redes de vários sectores sejam convergentes.

Como nós sabemos, existem actualmente diferentes ferramentas de gestão de redes, cada uma delas com funções específicas e independentes. Todas essas aplicações têm sido desenvolvidas para resolver alguns problemas que os fornecedores de serviços encontram no dia a dia na gestão e operação de redes.

Numa altura em que abundam os serviços disponíveis para utlizadores finais, e existem mais fornecedores de serviços a prestar os mesmos serviços, essas ferramentas de gestão não estão completas para responder a novos problemas de gestão de redes.

Neste contexto, e como resposta aos novos problemas de gestão de rede, pensamos que é importante criar um modelo que permite a integração de várias e diferentes ferramentas de gestão de rede. O modelo deve permitir que a informação fornecida por essas ferramentas de gestão de redes, seja guardada e partilhada para outras aplicações e redes, e até mesmo para gestores de redes.

A palavra chave, para este sistema de informação, é o serviço de directório, porque fornece um método standard, aberto e livre de guardar toda a informação necessária acerca das redes. Com este serviço podemos guardar a informação acerca dos objectos de rede, criar associações entre objectos, e utilizar a informação em aplicações autónomas.

Nesta tese descreve-se uma proposta de arquitectura de gestão de rede com o suporte de serviços de directório na partilha de informação de gestão de rede entre as diferentes aplicações de gestão para gestão autónoma. Para que essa gestão autónoma seja possivel, pensamos em reunir modelos existentes e esquematizar para uma possivel implementação com o recurso a directórios.

Resume

As the size of networks continue to grow, more and more network devices need to be managed efficiently, more network information must be managed, this takes to a situation that is quite impossible to perform a good management. Another aspect very important is that heterogeneous network technologies must co-exist and inter-work, in the way that infrastructures of network from many sectors converge.

As we know, different tools of network management currently exist, each one with specific and independent functions. All of these applications have been developed for solving that the service providers find in day-to-day network management and operation.

In a time when we see more and more services available for end users and more service providers to supply them, these management tools are not the complete answer to the new network management problems.

In this context and as an answer to new network management problems, we think that is important to create a model that allows the integration of diferent network management tools. The model must allow that the information supplied by those network management tools being stored and shared for others applications and networks, even to network managers.

The key component for this information system is the LDAP protocol, because it provides an standard, open and free method of storing all the information that we need about networks. With LDAP directories services, the information about network objects can be stored, associations between objects created, and information can be used in autonomous applications.

In this thesis is described a propose of network management architecture with the suport of directories services for share information of network management between diferent applications for autonomous management. For that management being possible, is necessary grouping existent models e make schemas for a possible implementation with the help of directory.

Résumè

Au fur et à mesure que la taille du réseau continue à grandir, multipliants les équipements de réseaux qui nécessitent d'être gérés efficientement, plus d'information de réseau doit être gérée, ce qui nous ramène à une situation presque impossible d'effectuer une bonne gestion. Une outre question très importante, c'est que les technologies de réseaux hétérogènes doivent coexister et inter opérer, dans la mesure que les infrastructures de réseaux de plusieurs secteurs convergent.

Comme nous le savons, il existe actuelment différents outils de gestion de réseaux, chacun eux avez des fonctions spécifiques et indépendantes. Toutes ces applications ont été développées pour résoudre quelque problème que les fournisseurs de services rencontrent chaque jour dans la gestion et opération de réseaux.

Dans un temps où abondent les services disponibles pour des utilisateurs finaux, et il existe plus de fournisseurs de services à prêter ces même services, ces outils de gestion ne sont pas complets pour répondre à de nouveaux problèmes de gestion de réseaux.

Dans ce contexte, et comme réponse aux nouveaux problèmes de gestion de réseaux, nous pensons qu'il é important créer un model qui permet l'intégration de plusieurs et différents outils de gestion de réseaux. Le model doit permettre que l'information fournie par ces outils de gestion de réseaux, soit gardée et partagée pour d'autres applications de réseaux, et aussi pour des gesteur de réseaux.

Le mot clef, pour ce système d'information, c'est le système de directoire LDAP parce qu'il fournie une méthode standard, ouvert et livre de garder toute l'information nécessaire sur les réseaux. Avec le service de directoire LDAP, l'information sur les objets de réseaux peut être garder, créer des association entre des objets, et utiliser l'information dans des applications autonomes.

Dans cette thèse est décrit une proposition d'architecture de gestion de réseau avec le support de services de directoire qui partage l'information de gestion de réseau entre différentes applications de gestion pour gestion

autonome. Pour que cette gestion autonome soit possible, nous avons pensé réunir des model existant e schématiser pour un possible implémentation avec le recours o directoires.

Agradecimentos

Na realização da tese foram nucleares os contributos de algumas instituições e algumas pessoas, a quem quero apresentar aqui os meus sinceros agradecimentos.

Agradeço à Faculdade de Engenharia da Universidade do Porto, por ter aceite a candidatura do meu trabalho, o qual espero venha a ser um valioso contributo para a comunidade académica desta Universidade, constituindo a base para futuros trabalhos de investigação;

Agradeço à Engenheira Helena Machado, minha esposa, pela importante colaboração e compreensão que me forneceu na superação de dificuldades e pelo incessante apoio;

Agradeço imenso ás pessoas amigas que me ajudaram na realização desta tese. Queria tambem aqui expressar, a outras pessoas (sem querer mencionar nomes, pois elas sabem a quem me estou a referir), o meu dobrado estado de desejo assim como me desejaram;

Agradeço imenso a minha **mãe** pelos seus sacrifícios e incentivo.

Nomenclatura

ACPI - Advanced Configuration and Power Interface
CIM - Common Information Model
DEN - Directory Enabled Network
SNMP - Simple Network Management Protocol
CMIP - Common Management Information Protocol
CMISE - Common Management Information Service Element
DMI - Desktop Management Interface
OSI - Open Source Initiative
MIS - Management Information Structure
MIB - Management Information Base
GDMO - Guidelines for Definition of Managed Objects
ASN.1 - Abstract Syntax Notation One
TCP - Transmission Control Protocol
IP - Internet Protocol
UDP - User Datagram Protocol
ISO - International Organization for Standardization
LDAP - Lightweight Directory Access Protocol
ITU - International Telecomunication Union
TMN - Telecommunications Management Network
SMI - Structure of Management Information

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

O rápido crescimento da Internet durante os últimos anos, criaram a necessidade de serviços de directório mais robustos, escaláveis e seguros.

Um dos grandes desafios que se pôem à automatização de tarefas de gestão em redes de telecomunicações é a supressão do factor humano através da introdução de entidades de software que colhem, processam e organizam informação, e possuem uma habilidade inata para actuarem nos seus equipamentos de controlo, de comutação e de encaminhamento. Este é o modelo de gestão emergente de estudos recentemente desenvolvidos.

Um serviço de directório é a tecnologia de suporte ideal para a criação e modelização de redes "inteligentes". Este serviço é fisicamente distribuido e um repositório de dados lógicamente centralizados que alteram poucas vezes num ambiente computacional.

As redes têm-se tornado muito complexas e com grande variedade de equipamentos de rede existentes, em que cada elemento executa diferentes protocolos e serviços sobre vários meios, disto resulta que existam várias aplicações especificas de gestão de redes, sem a possibilidade de inter-agirem uns com os outros. Assim o serviço de directório oferece a promessa de ser um repositório central para guardar e consultar informação.

1.1 Área de Intervenção

Este trabalho de investigação insere-se numa proposta de criação e optimização de modelos de gestão, na área de redes informáticas, e baseia-se no recurso ao serviço de directório para o suporte á criação basilar de uma rede "inteligente". Uma boa gestão dos equipamentos informáticos e de redes é essêncial para o bom funcionamento do ambiente computacional.

O objectivo da tese é investigar a possibilidade de integrar a informa-

ção de gestão obtida, a partir das aplicações de gestão de redes, e colocar essa informação num sistema fiável, comum e simples, de onde as aplicações, os equipamentos e até mesmo os utilizadores podem tirar partido dessa informação integrada, usando por isso o serviço de directório, ao invés das tradicionais base de dados proprietárias, e ficheiros de configuração.

1.2 Serviço de Directório e Gestão de Redes

Normalmente os equipamentos de rede têm dois estados, estado dinâmico e o estático. O estado dinâmico é bem suportado pelos protocolos de gestão de rede, contudo não existe nenhum standard que defina onde e como guardar o estado persistente.

A integração da infra-estrutura de rede com o serviço de directórios permite que as aplicações e os utilizadores descubram a existência de relações ao interrogar o serviço de directório. Assim deste modo é mais escalável e gerivel do que contactar cada equipamento individualmente e agregar os resultados. Colocar os equipamentos de rede num directório aumenta a sua gestão e utilidade, enquanto reduz o tráfego na rede.

São já conhecidas as grandes vantagens do serviço de directório para inserção, alteração, remoção, pesquisa e extracção de informação ou dados. Recomenda-se a leitura de dois livros muito bons nesta matéria, na qual faço referência na Bibliografia ([13] e [15]), o qual se revela útil para quem pretende iniciar o seu estudo no serviço de directório.

1.2.1 Conhecidos Beneficios de um Directório

Tradicionalmente um directório fornecia um meio para localizar e identificar utilizadores e recursos disponiveis num ambiente de rede. Tambem fornecia a possibilidade de adicionar, remover, alterar, renomear e gerir elementos sem interromper os serviços fornecidos pelos outros equipamentos.

[12]Hoje em dia um directório é usado da seguinte forma:

- simplificar a administração de redes;
  - gestão centralizada da informação das pessoas;
  - gestão centralizada da configuração de computadores e máquinas;
  - gestão centralizada das contas dos utilizadores;
  - redução de custos através da gestão centralizada;
- unificar acesso a recursos de rede;

  - único "login" para aceder aos recursos da rede;
  - convenção de nomes uniforme;

- fornecer um caminho aos utilizadores para pesquisarem informação;
  - localização centralizada dos recursos da rede;
  - serve de catálogo para qualquer tipo de dados, por exemplo, documentação de produtos;
  - informação de contacto;

- melhorar a gestão de dados;
  - melhora a consistência dos dados que são muito usados;
  - fornece segurança para dados críticos;
  - organiza os dados numa estrutura lógica;

- fornecer capacidade de repositório e busca de informação para aplicações e serviços de dados;

Alguns destes benefícios do uso do serviço de directórios estão a ser aproveitados em larga escala e em muitas aplicações, o qual é feita referência no Capítulo 2 desta tese.

1.3 Projecto e Parceria

Este trabalho integra-se num projecto que está a ser desenvolvido na FEUP[1] em parceria com a HP[2], com o fornecimento da aplicação de gestão de redes "HP Node Manager".

Neste momento, todas as tabelas da base de dados que serviam para guardar informação dos equipamentos e topologias de redes da aplicação "HP Node Manager", estão mapeadas num directório, sendo por isso, um grande ponto de partida na possibilidade de migração para outro tipo de sistema de informação comum.

Outra tarefa que está a ser elaborada em directório no grupo, consiste na implementação de um repositório de informação para os alertas (eventos) dos equipamentos e encontrar uma maneira de os mostrar e tratar.

Com a elaboração desta tese, pretende-se ir um pouco mais longe, encontrar modelos que possibilitem uma melhor solução de gestão e resolução de problemas que possam surgir numa rede de próxima geração.

[1] Faculdade de Engenharia da Universidade do Porto
[2] Hewllet Packard

1.4 Problema a resolver

O desenvolvimento do serviço de directório e as suas grandes potencialidades, tornam pertinentes a utilização deste benéfico serviço para a comunidade informática na gestão de redes.

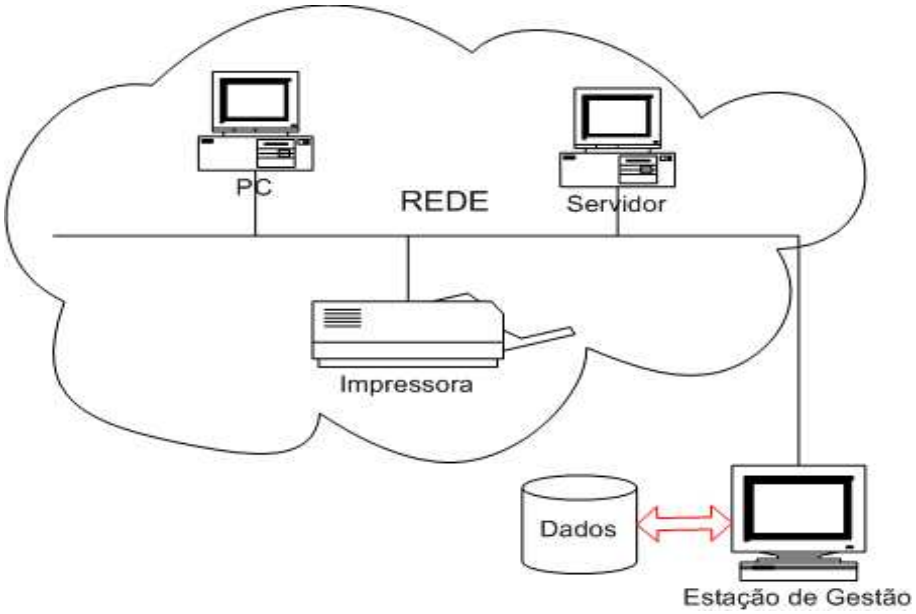

Figura 1.1: Modelo Actual na Gestão de Redes

Um dos problemas actualmente existente nas aplicações de gestão de redes prende-se com o facto de não terem sistemas de informação normalizados para publicar ou disponibilizar informação entre aplicações de gestão. Outro problema inerente, é serem aplicações fechadas e proprietárias, tornando difícil a percepção e análise dos dados que, normalmente, são guardados em Bases de Dados, também estas proprietárias.

1.5 Abordagem para sua Resolução

Para solucionar este problema, recorreu-se ao uso do serviço de directório LDAP[3], pois este sistema de informação tem imensas capacidades para repetitivas leituras e pesquisas, dar resposta rápida e eficaz em alternativa às usuais Bases de Dados.

[3]Lightweight Directory Access Protocol

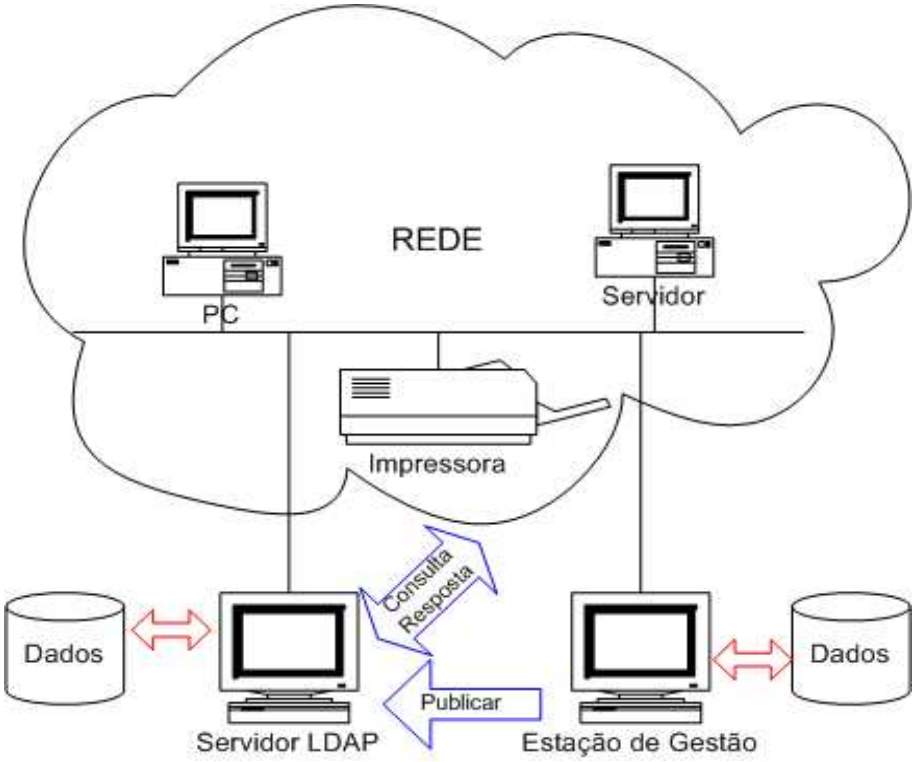

Figura 1.2: Modelo Pretendido para Gestão de Redes

Para que se consiga um modelo de rede "inteligente" é necessário dotar e escolher o suporte de informação que seja possivel o seguinte:

- fornecer modelo para guardar informação dos elementos de rede e serviços (árvore Park Tree);
- possivel mapear (fisica e lógicamente) topologias de redes (árvore Discovery Tree);
- descrever como funcionam os elementos de rede (árvore Containment Tree);

Com o recurso do serviço de directório para fornecer suporte da informação atrás referida, tira-se partido e significativos beneficios para as aplicações e gestores de redes, como sejam:

- gestão de inventário actualizado;
- melhora a gestão dos equipamentos;

- configuração e fornecimento de dados às aplicações;
- capacidade de planear;

Esta tese propõe ainda a utilização de um conjunto de definições e de modelos abstractos (CIM[4] e DEN[5]), as quais tencionam ser standards para gestão de redes, fazendo para tal, o uso de um serviço de directório LDAP que é capaz de dar resposta eficiente ao pretendido.

1.6 Estrutura da Tese

A tese é composta por quatro capítulos, sendo o primeiro a **Introdução**.

No **Capítulo 2** aludem-se algumas considerações e referências ao estado actual do uso de LDAP num ambiente computacional.

No **Capítulo 3** faz-se referência e modelização de possíveis modelos de informação, para ser implementado e realizado num serviço de directório LDAP, para atingir um grau primário de rede inteligente.

No **Capítulo 4** apresenta-se a Conclusão, onde consta a referência conclusiva acerca do trabalho desta tese e dá a conhecer as potencialidades no futuro desenvolvimento de sofware de gestão de redes.

A seguir enuncia-se a **Bibilografia** onde se explicítam alguns documentos de suporte para a realização desta tese, sendo alguns livros, outros documentos disponíveis na Internet, e ainda endereços onde reside informação de extrema utilidade.

No fim da tese surge o **Apêndice**, onde é exposto o código que serviu de base para algumas experiências, e a forma como foram implementados os modelos referênciados na tese.

[4] Commom Information Model
[5] Directory-Enabled Network

Capítulo 2

Estado da Arte do LDAP

2.1 Introdução

O conceito do uso de serviços de directório, em ambiente de redes computacionais, não é novidade. Muitas empresas implementaram e, ainda continuam a usar este serviço de directórios, como suporte básico de informação acerca dos seus empregados (por exemplo, endereço da residência, endereço de e-mail e número de telefone, etc) disponível para toda a empresa. A autenticação e autorização dos utilizadores para o uso dos recursos informáticos da empresa são muitas vezes feitos com base na informação contida em directório.

Neste capitulo faz-se uma breve alusão a algumas aplicações de serviços de rede, na qual o serviço de directório LDAP está, e em vias de ser implementado. Não é objectivo deste capitulo mostrar todos, mas sim algumas aplicações de grande importância, e demonstrar a sua grande valência e flexibilidade em vários tipos de serviços, num único directório LDAP.

2.2 LDAP nos Sistemas Operativos

O LDAP, actualmente, é usado por uma imensa variedade de fabricantes de aplicações. A título de exemplo a Novell mostrou em 1993, com o lançamento do NetWare, que é possivel simplificar a gestão de tarefas básicas ao incluir um serviço de directório (o NDS[1]) como parte integrante da sua oferta. Depois, seguiu-se a Microsoft, no ano 2000, com o lançamento do "Windows 2000 Server" possuindo a funcionalidade de serviço de directório AD[2] para fins próprios de autenticação de utilizadores, de políticas, e gestão de máquinas (PCs, Servidores e Impressoras) no seu dominio e tendo, por

[1]Novell Directory Server
[2]Active Directory

isso, já alguma notoriedade na utilização deste serviço. A IBM, a SUN, a HP, a Apple e a comunidade Linux têm feito esforços para incluir o suporte a serviços de directório nas suas distribuições de sistemas operativos.

2.3 Servidores de Email e LDAP

Alguns servidores de e-mail fornecem serviços de directório na forma de agenda de endereços para simplificar o processo de pesquisa de contactos e enviar um e-mail. Esta é a razão principal de hoje em dia existir muita aplicabilidade deste serviço, sendo, por isso, muito utilizado como complemento de servidores de e-mail. Muitas aplicações hoje em dia tiram partido do LDAP, explicitando aqui apenas algumas mais usuais:

- QMail;
- Sendmail;
- Postfix;
- MS Exchange;

2.4 RADIUS e LDAP

É também visivel o uso do LDAP para guardar e fornecer informação para o RADIUS[3], em substituição dos tradicionais ficheiros de configuração e de informação desse mesmo serviço. Pode-se consultar algum desenvolvimento de esquemas LDAP para o uso nas seguintes aplicações:

- Cistron RADIUS;
- FreeRADIUS;
- OpenRADIUS;
- XtRADIUS;

Estes são apenas algumas aplicações que fazem uso do directorio LDAP para guardar a respectiva informação.

[3]Remote Authentication Dial-In User Service

2.5 DHCP e LDAP

É de bom grado que toda a informação de rede esteja focalizada num sitio com a funcionalidade de ser distribuida e partilhada, ou pelo menos parte dela, e o serviço de DHCP não foge á regra. Qualquer administrador de redes gosta de ter a máxima informação possivel na sua árvore LDAP.

É possivel ver a disponibilidade e funcionalidades do DHCP obter informação do directorio, para as máquinas na rede que necessitem dessa informação. Como é do conhecimento de toda a comunidade informática, o DHCP é um protocolo de distribuição de endereços IP para o normal funcionamento em ambiente de rede. O respectivo esquema já foi desenvolvido e está disponivel na Internet para quem quiser usar e experimentar.

2.6 Serviço de Resolução e LDAP

Num ambiente de rede, cada host necessita de informação acerca da rede e de outros hosts para poderem comunicar. Existem uma variedade de mecanismos para resolver nomes numa rede, e esses mecanismos são:

- DNS - fornece a resolução host/IP e vice-versa;
- NIS/NIS+ - fornece uma base de dados centralizada para serviços de informação comuns dos utilizadores, grupos, hosts, redes, protocolos, serviços, RPC, etc;
- Ficheiros locais /etc, onde reside a informação em cada host;
- e LDAP - Nova adição á familia, fornece informação acerca de utilizadores, grupos e hosts usando o LDAP;

O serviço de resolução de nomes fornece um mecanismo centralizado e de distribuição de informação aos hosts da rede, eliminando a necessidade de configurar essa informação localmente em cada host, reduzindo drasticamente o esforço de dirigir uma rede.

2.6.1 DNS/DNSsec e LDAP

Inicialmete a informação de DNS era guardada em simples ficheiros de texto, sendo muitas vezes desejável guardar informação de DNS numa base de dados devido a facilidades na redução de informação repetida. A natureza da informação de DNS é hierarquica e ao usar uma base de dados relacional não é eficaz, porque torna-se dificil guardar informação que seja de cariz livre.

Aqui o LDAP tem um papel preponderante pela sua capacidade de fornecer esquemas de informação hierarquica e múltipla, por exemplo, o administrador pode definir mais do que um IP para cada nome canónico sem quebrar as regras de normalização.

Algumas aplicações de DNS oferecem suporte para LDAP como sejam:

- ISC Bind (Named);
- TinyDNS;
- PowerDNS;

O RFC2247 é uma versão draft, que define o esquema de como usar a informação DNS em LDAP, e que na qual os servidores DNS atrás mencionados não suportam toda a funcionalidade, mas sim parte do RFC mencionado.

Hoje em dia fala-se muito em segurança e o DNS tem algumas falhas de segurança de autenticidade, daí existir uma versão de DNS com funcionalidades extra de segurança, e existe mais valia na possibilidade de guardar informação de DNSsec em LDAP e estar disponivel essa informação para a resolução de nomes.

2.6.2 NIS/NIS+ e LDAP

Uma das formas de oferecer o serviço do NIS/NIS+ é recorrer ao suporte LDAP, e está referênciado no RFC 2307. O NIS/NIS+ permite que os seus clientes usem de forma transparente o LDAP para resolver utilizadores, grupos, redes, serviços, informação das máquinas, etc.[22] O RFC2307 define e emprega o esquema para "Using LDAP as a NIS", sendo este esquema abarcado por vários fabricantes incluindo a Apple, HP, Novell, Red Hat, Silicon Graphics e Sun Microsystems.[21]

2.7 VoIP e LDAP

Está a decorrer uma investigação e sua elaboração de tese, na FEUP, de como é possivel desenvolver um proxy VoIP[4] recorrendo ao uso do LDAP. O objectivo deste sistema de informação consiste em guardar registos de utilizadores de VoIP, para sua posterior obtenção de informação vital para estabelecer a comunicação entre dois ou mais interveninetes registados no servidor LDAP.

[4] Voice over IP

2.8 PGP/CertServ e LDAP

É tambem possivel e existe algumas experiências na distribuição de chaves públicas e privadas (PKI) assim como de certificados (X.509), recorrendo ao uso de directórios. É de extrema utilidade um serviço de distribuição de informação para implementar um certo grau de confidencialidade e segurança de informação que percorre nas redes. Assim é de vital importancia a utilidade da implementação de informação básica para "cifrar" a informação entre os emissores e os receptores dessa informação.

2.9 Linguagens de Programação e LDAP

Um dos mais importantes aspectos do desenvolvimento do LDAP, é que ele é um protocolo simples e é relativamente simples de implementar e trabalhar. Isto fica transparente pelo facto que o LDAP é suportado pela maior parte das linguagens de programação, incluindo o C, Java, Perl e aplicações de páginas dinamicas (como o PHP) e está a ser suportada pela maior parte dos sistemas operativos, como atrás foi referido neste capitulo. Isto torna fácil a integração dos programadores e valor acrescentado para futuros desenvolvimentos de aplicações.

2.10 Caso de Estudo: Novis e M-Vault

A NOVIS é uma empresa Portuguesa de Telecomunicações e subsidiária de SonaeCom. A SonaeCom tem empresas que fornecem serviços móveis (Optimus), serviços de Internet (Clix) e serviços de Multimedia (Matrix) na qual a NOVIS presta e fornece serviços ás empresas da holding.

Antes de usar o M-Vault (serviço de directório da ISODE), a NOVIS tinha várias bases de dados para cada serviço, usando DB2, DB3 e CDB. O uso de bases de dados relacionais foi considerado muito complexo e havia grandes problemas de performance em pesquisas de informação em larga escala.

O LDAP parecia ser uma solução standard de alta performance e que fornecia requisitos para os serviços. O M-Vault da Isode foi escolhida como fornecedora de directorio LDAP. A operação inicial contava com 500 000 entradas (registos), que rapidamente aumentou para 2 milhões em Janeiro de 2003. O porquê desta mudança e escolha prendeu-se fundamentalmente em 4 pontos:[20]

- o servidor LDAP é compativel com os standards;

- é extremamente eficiente;
- é escalável sem custos;
- é fácil a sua instalação e manutenção;

A NOVIS usa o M-Vault para dar suporte ás seguintes aplicações:

- Entrega de EMail e Autenticação;
- Serviço de Hosting, para permitir que os clientes construam e publiquem as suas páginas;
- Acesso á Internet
  - Dial-Up usando o RADIUS[5];
  - DSL[6];
- Autenticação via serviço WEB;
- Autenticação dos utilizadores internos para acesso aos servidores;

2.11 Conclusão

Neste capitulo fez-se uma pequena alusão das potencialidades e flexibilidade do uso do serviço de directório nos mais variados serviços que podem existir numa rede. A capacidade de guardar vários tipos de informação de forma hierárquica para aplicações, utilizadores, e até mesmo definir politicas para os recursos existentes na rede. Não foram aqui divulgadas todas as propostas em uso ou investigação, tendo em atenção que muito está para ser explorado e divulgado. Sem duvida que se vê grande capacidade e flexibilidade do uso do LDAP como base de dados de configurações e informações de vários serviços aplicacionais como de redes.

[5] Remote Authentication Dial-In User Service
[6] Digital Subscriber Line

Capítulo 3

Proposta e Âmbito da Tese

3.1 Introdução

Pretende-se com esta tese demonstrar a existência de meios e funcionalidades para que se possa integrar informação de rede com o serviço de directório.

Isto é uma iniciativa requerida pelos fabricantes de equipamentos, fabricantes de software independendentes e utilizadores, para definir um modelo racional e utilizável para melhorar a capacidade de gestão de redes com a integração do serviço de directório. Os recursos de rede (equipamentos, sistema operativo, ferramentas de gestão e aplicações) usam o serviço de directório para:

- publicar informação acerca deles próprios;
- descobrir outros recursos;
- obter informação acerca de outros recursos;

A convergência para guardar informação de gestão tem sido dificil. A falta de integração e a pura complexidade das ferramentas, são por si próprias uma barreira ao desenvolvimento de novas aplicações, assim como na troca e partilha de informação de dados obtidos por aplicações individuais de gestão.

Os administradores necessitam de um nivel de controlo sobre as suas redes, que actualmente não está disponivel.

Um serviço de directório escalável e seguro que apresente uma vista logicamente centralizada de informação fisicamente distribuida, é o serviço adequado para guardar a meta-informação essencial para criar e gerir uma rede da próxima geração ("rede inteligente"). A especificação para a integração dos serviços de directório e os serviços de rede, definem o modelo de informação e esquemas que tornem isso possivel.

3.2 Porquê da escolha do LDAP

O LDAP foi projectado específicamente para resolver os problemas causados pela proliferação de diretórios pela rede, e existem sete aspectos de sua implementação actual que lhe dão esta habilidade.

Desenho genérico - Como o LDAP foi projectado para ser um diretório de propósito geral, ele teve que ser extensível. Ele usa um esquema de definição orientado a objectos, baseado em herança, que permite fácil extensão para qualquer uso razoável. Existe um esquema básico que é parte da especificação do LDAP, e existem outros padrões de facto para vários serviços. Entretanto, espera-se que a maioria dos programadores irá extender os esquemas básicos.

Simplicidade do protocolo - Um dos mais importantes aspectos do desenvolvimento do LDAP, e que fez com que o mesmo fosse adoptado em vez do DAP, é que ele é um protocolo simples e é relativamente fácil de implementar e trabalhar. Isto é transparente pelo facto que o LDAP é suportado pela maior parte das linguagens de programação, incluindo C, Java e Perl, e ou é suportado ou será suportado pela maior parte dos sistemas operativos, incluindo o Solaris, GNU/Linux, Microsoft Windows, e o Mac OS.

Arquitetura distribuída - Com o uso de replicação de dados, é possível replicar todo ou parte de um diretório LDAP para locais separados fisicamente, o que permite que os dados tenham alta disponibilidade e coloca os mesmos tão próximos quanto necessário do cliente. Utilizando referências, domínio de dados de porções do diretório podem ser distribuídos em diferentes servidores LDAP, permitindo assim que partes de uma empresa ou projeto tenham controle sobre os dados necessários ao mesmo tempo que mantém uma única autoridade sobre cada parte dos dados.

Segurança - Um grande foco do desenvolvimento do LDAP foi a segurança, com a versão 3 do protocolo LDAP trazendo significativos melhoramentos. Existem três aspectos básicos na proteção de informação em um diretório: Acesso, Autenticação e Autorização (AAA, ou Triplo-A). **Acesso** é a habilidade de conectar-se a um serviço e pode ser restringida baseado em detalhes como hora do dia ou endereço IP, **Autenticação** é a habilidade de provar ao serviço que um cliente é um utilizador válido, e **Autorização** é o serviço fornecendo ou negando direitos específicos ou funcionalidades ao cliente. Infelizmente, a sintaxe das ACLs

ainda não é parte da especificação LDAP. Parece que a implementação da Netscape das ACLs será aceita como padrão, mas isto ainda não aconteceu e diferentes servidores LDAP implementam as ACLs de diferentes formas. Entretanto, isto não deve afectar o desenvolvimento ou funcionamento dos clientes. Para acesso seguro, o LDAP suporta o Transport Layer Security (TLS), que criptografa toda a comunicação entre cliente e servidor. Para autenticação, o LDAP suporta a Simple Authentication and Security Layer (SASL), que permite que o cliente e servidor negociem um método de autenticação (seguro). O TLS e o SASL provêm funcionalidades criptográficas mas não o controle sobre o acesso e autenticação. O LDAP irá fornecer a habilidade de controlar todos os três aspectos da AAA através de Access Control Lists (ACLs). As ACLs podem ser usadas para autorizar o acesso baseado em muitos fatores diferentes. Elas podem ser usadas para forçar tipos específicos de autenticação e, uma vez que o cliente esteja plenamente autenticado como utilizador válido, as ACLs são usadas para autorizar o utilizador.

Padrão aberto - Como o LDAP é um padrão aberto mantido pela IETF, ele pode ser utilizado por qualquer programador, companhia, ou administrador sem receio de ficar preso a protocolos proprietários ou a fabricantes específicos, e permite que a escolha da implementação seja baseada nos detalhes do projeto em vez de preocupações de interoperabilidade. Isto também significa que o LDAP pode progredir de acordo com as necessidades das pessoas que o utilizam, em vez de uma corporação que está concentrada nos lucros ou fatia de mercado.

Solicitação de funcionalidades e esquemas do servidor - A especificação LDAP garante que os clientes LDAP podem solicitar a lista completa de funcionalidades de esquemas de dados de qualquer servidor LDAP, permitindo desta forma que o cliente altere sua funcionalidade de acordo com a do servidor, que deverá fornecer grande interoperabilidade entre diferentes implementações e versões do LDAP.

Internacionalização - O LDAP utiliza o UTF-8 para representação interna de strings. Isto permite que o LDAP armazene e manipule qualquer linguagem do mundo.

Esta não é uma lista exaustiva de todas as funcionalidades do LDAP, mas detalha alguns dos mais importantes aspectos do protocolo. De facto, uma das razões pela qual o LDAP está sendo adoptado agora é apenas marginalmente relacionada ao LDAP em si: a indústria de computação, especialmente na área de administração de redes, está pronta para diretórios corporativos.

Isto é evidenciado pelo fato destes estarem surgindo em todos os cantos, de servidores como o Directory Server da Netscape e o Active Directory da Microsoft a clientes como clientes de email e sistemas operacionais todos atendendo a especificações padrão como a iniciativa DEN[1].

3.2.1 Utilidade do CIM e DEN

O CIM[2] é um modelo conceptual orientado a objectos para informação requerida a gerir muitos e comuns aspectos de sistemas computacionais complexos definidos pelo DMTF. O objectivo do CIM é a apresentação de um modelo consistente de ambientes independentes geriveis de vários protocolos e formatos de dados suportados pelos equipamentos e aplicações. O CIM tem ganho aceitação como um modelo de informação para ferramentas de gestão empresariais de vários fabricantes de software e hardware. Em particular, o uso do CIM promove sinergias entre integração, melhorias da rede e aplicações de directório e gestão que usam o CIM

A representação do DEN na informação dos elementos de rede complementa e melhora o modelo actual do CIM, isto é, o DEN extende a actual versão do CIM. O esquema do DEN incorpora conceitos do X.500 e do CIM, e deve ser visto como um veiculo de junção de diversos tipos de informação juntos.

3.3 Visão da Tese

A visão deste trabalho é reunir informação que permita duas coisas:

1. ligar os utilizadores aos recursos existentes na rede;

2. fornecer a base para criar redes e aplicações inteligentes;

Estes dois pontos requerem aplicações que possa ter acesso á diversa informação a partir de um repositório comum. Este repositório usa o serviço de directório como suporte de informação centralizada e que coordene as funções de guardar e consulta de informação, e permita que outros dados e aplicações sejam unificadas.

[1] Directory Enabled Networks
[2] Common Information Model

3.4 Solução do Problema

Para implementação da solução do problema, referênciado no Capítulo 1, com os objectivos para uma gestão de redes eficiente e "inteligente", com um certo grau de automatização e acções pró-activas, foi criado como base três ramos distinctos para os diferentes tipos de informação que pretendemos guardar, ou seja, guardar toda a informação sobre o parque informático (PT[3]), a topologia da rede (DT[4]) e as regras de conectividade (CT[5]). As árvores de CT, PT e DT contêm informação de extrema importância para a resolução de problemas que possam surgir na rede, e ajudam o gestor de redes e as aplicações de gestão a tomar decisões (os problemas são descobertos com a ajuda de alertas).

As estruturas, dos três ramos definidos, serão úteis para as aplicações "inteligentes" de gestão de redes e aos gestores de redes, de forma a obterem informação sobre a rede e o seu estado, dos equipamentos contidos na rede e as regras de ligação dos equipamentos.

No Capítulo de Apêndice foram incluidas algumas imagens (snapshots) de um browser LDAP acerca destas três árvores que aqui se relata, na qual dá uma visão inicial do propósito pretendido com este trabalho. É de salientar que a implementação futura destes modelos não é obrigatória seguir esta hirarquia no DIT do serviço de directório, sendo livre para qualquer pessoa ou organização que queira iniciar e implementar estes modelos.

3.5 Árvore de Conteúdo

3.5.1 Objectivo

O objectivo desta secção é mostrar o modo como se pode mapear em LDAP as ligações que cada equipamento de rede pode ter, isto é, saber quais as possibilidades de ligações que cada equipamento inserido na rede pode ter. Este modelo serve para ajudar na descoberta dos equipamentos de rede, fazendo para isso regras de conexão entre os equipamentos, sendo criada uma árvore CT[6] para o efeito pretendido.

A aplicação deste modelo na árvore DIT[7], tem por objectivo prestar uma significativa ajuda na descoberta de problemas na rede, no caso de um equi-

[3]Park Tree == Árvore Parque Informático
[4]Discovery Tree == Árvore de Descoberta
[5]Containment Tree == Árvore de Conteúdo
[6]Containment Tree == o mesmo que Árvore de Conteúdo
[7]Directory Information Tree

pamento não estar no lugar devido ou a sua conectividade não seguir regras, assim como na descoberta da topologia de rede e descobrimento dos respectivos elementos inseridos na rede.

3.5.2 Arquitectura CT

Cada contentor é definido como sendo uma gama ou grupo de equipamentos de rede, isto é, existem grupos de PCs, HUBs, SWITCHs, ROUTERs, PRINTERs, etc, no qual cada equipamento encontrado na rede pertence a um destes grupos.

O esquema apresentado na Figura 3.1 é o mais simples de implementar e aquele que não necessita de muitos contentores relativamente ao referênciado na Figura 3.2. Cada classe contêm atributos na qual se define que gamas ou grupos de equipamentos podem conectar-se a eles.

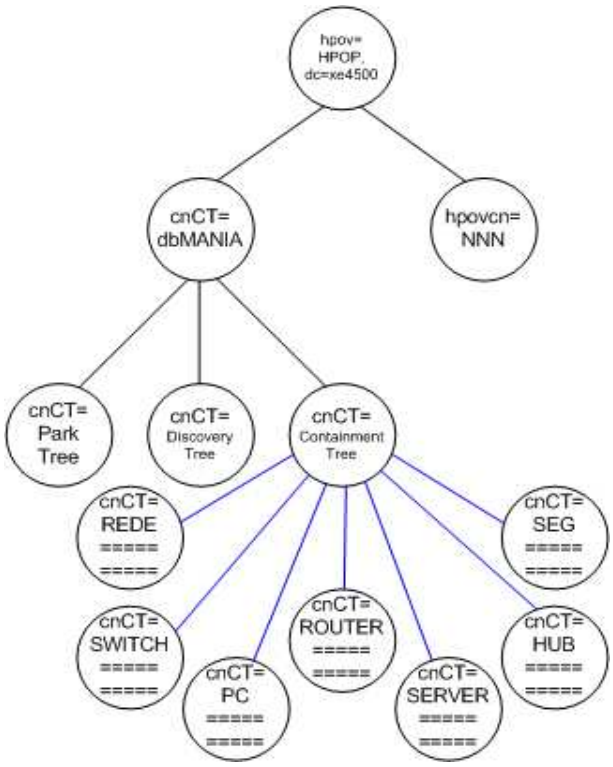

Figura 3.1: Estrutura Funcional do Containment Tree - versão 1

Uma outra alternativa é a que se apresenta na Figura 3.2. É uma árvore DIT completa com todos os contentores LDAP, dando um aspecto global

de como irá ficar a árvore, contendo toda a informação necessária para a descoberta dos equipamentos de rede.

É de salientar, que em ambos os esquemas, também os equipamentos que não fazem parte directa da rede possuem regras de conexão. São aqueles que ficam conectados por outras interfaces (USB, LPT, COM, etc...), sendo esses equipamentos algumas impressoras, faxes, scaners, etc.

Toda a informação que tiver relacionamento com a "Árvore de Conteúdo" deve ficar por baixo do contentor LDAP "Containment Tree" ou "CT".

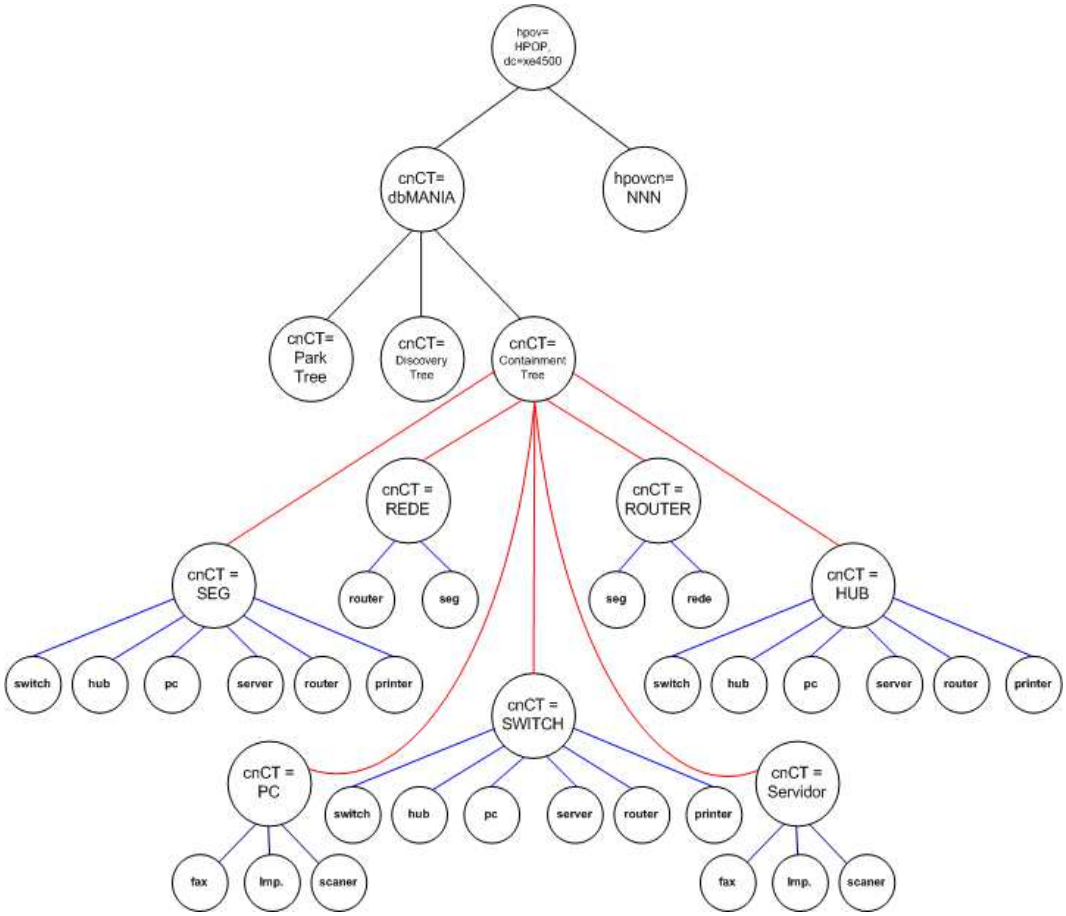

Figura 3.2: Estrutura Funcional do Containment Tree - versão 2

Para que este modelo funcione plenamente, é necessário que se faça uma colecção de possíveis contentores LDAP e se construam as possíveis ligações a outros equipamentos, para além de outras funcionalidades.

Para os equipamentos de rede finais (como alguns Faxes, algumas Impressoras, e alguns Scaners, etc...) ligados a contentores LDAP, não é neces-

sário criar regras de ligação, porque são equipamentos terminais na árvore de LDAP. Embora, estes equipamentos não estejam ligados directamente na rede, eles têm de estar conectados a equipamentos que lhes fornecem os serviços (como por exemplo, PCs e Servidores) através de serviços de redes.

3.5.3 Proposta em LDAP para CT

Como se pode verificar pela Figura E.1 incluida em Apêndice, como fica organizada a informação relativa a "Containmente Tree". Assim desta forma a árvore LDAP de CT fica simples e define bem as regras de conectividade entre os equipamentos existentes numa rede, estando essas possibilidades definidas nos seus atributos os possiveis grupos de equipamentos.

Foi definido o esquema exposto na Figura 3.1 na árvore de LDAP por ser a mais simples e por fornecer melhor conveniência para visualização da informação, não sendo necessário descer na árvore para tomar conhecimento de quais os equipamentos que podem estar conectados (regras).

3.6 Árvore de Descoberta

3.6.1 Objectivo

Neste capítulo faz-se a demonstração de um modelo de "Discovery Tree"[8] baseado em LDAP. O intuito deste modelo é mapear a estrutura de uma rede com todos os seus equipamentos (PCs, HUBs, SWITCHs, ROUTERs, etc...), os seus segmentos de rede, fazendo para isso uso de software ou aplicação de gestão de redes a fim de descobrir a topologia da rede pretendida.

Usando este modelo de mapeamento de redes em LDAP, outras aplicações de gestão de redes podem tirar partido da informação inserida no LDAP, tornando, desta forma, fácil e transparente o desenvolvimento de futuras aplicações compreensíveis até para o gestor de redes.

3.6.2 Arquitectura DT

Nesta secção demonstram-se com recurso de cenários simples, exemplos de redes para que se compreenda o modo de mapear uma rede e seus equipamentos de rede, seguindo as regras referidas no capítulo anterior (usando a árvore de CT[9]).

Cenário 1

A topologia de rede, explícita na Figura 3.3 é o primeiro exemplo simples. Estão directamente ligados à rede dois concentradores (um Hub e um Switch), oito PCs, uma Impressora, e dois Servidores (um servidor de ficheiros e outro de correio). Indirectamente ligados à rede estão um fax, um scaner e uma impressora.

É de salientar que um dos ramos ligado ao Hub1 é do tipo BNC (coaxial) ao qual estão pendurados quatro PCs nesse ramo. De seguida analisa-se o modo como ficam esses PCs mapeados na árvore LDAP. Por fim, verifica-se a simplicidade desta forma de mapeamento.

Na Figura 3.4 apresenta-se uma possível árvore DIT em LDAP para a topologia de rede explícita na Figura 3.3. Na descoberta dos equipamentos é de salientar que não foi encontrado nenhum Router, e, por conseguinte, iniciando-se então a pesquisa pelo Hub1, fazendo, por isso, o contentor LDAP

[8] Discovery Tree == DT == o mesmo que Árvore de Descoberta
[9] CT == Containment Tree

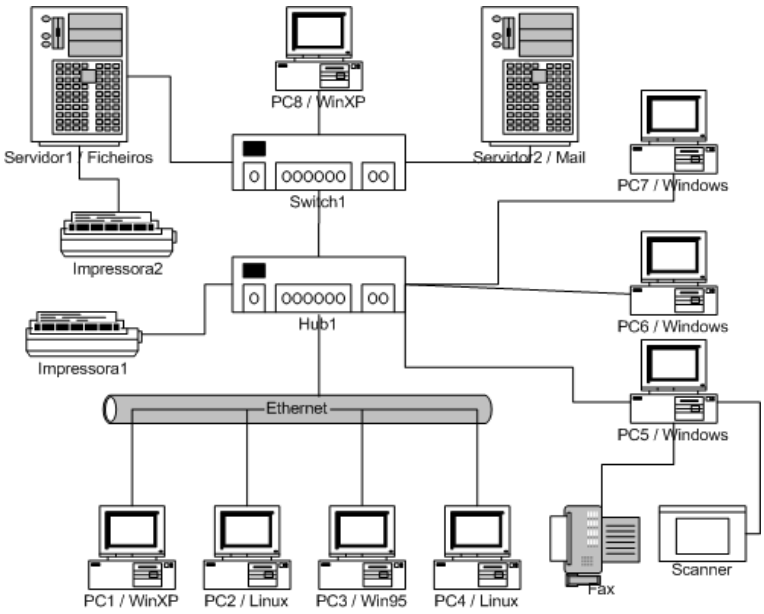

Figura 3.3: Arquetipo Rede - Cenário 1

"Segmento" ficar imediatamente ligado ao contentor LDAP **"Rede"** (seguindo a regra definida no capítulo anterior, a árvore de CT, as possibilidades que o contentor "Rede" tem para ligar são a um Router ou a um Segmento).

Cenário 2

A topologia de rede explícita na Figura 3.5 é o segundo exemplo simples. O cenário é idêntico ao exemplo anterior, no qual foi inserido um Router na topologia.

Doravante apresentam-se exemplos de alguma complexidade, com a introdução de equipamentos ou mudanças de topologia para tornar mais visível o objectivo proposto.

Com a introdução do router na estrutura da rede verifica-se que a árvore fica um pouco mais aberta (com mais ramos), onde estão pendurados todos os equipamentos nesses mesmos ramos do router (PCs, SWITCHS, etc...). Os ramos (segmentos da rede) são as possíveis rotas que estão contidas na tabela de encaminhamento do router, e consequêntemente são mapeadas em LDAP como sendo segmentos da rede.

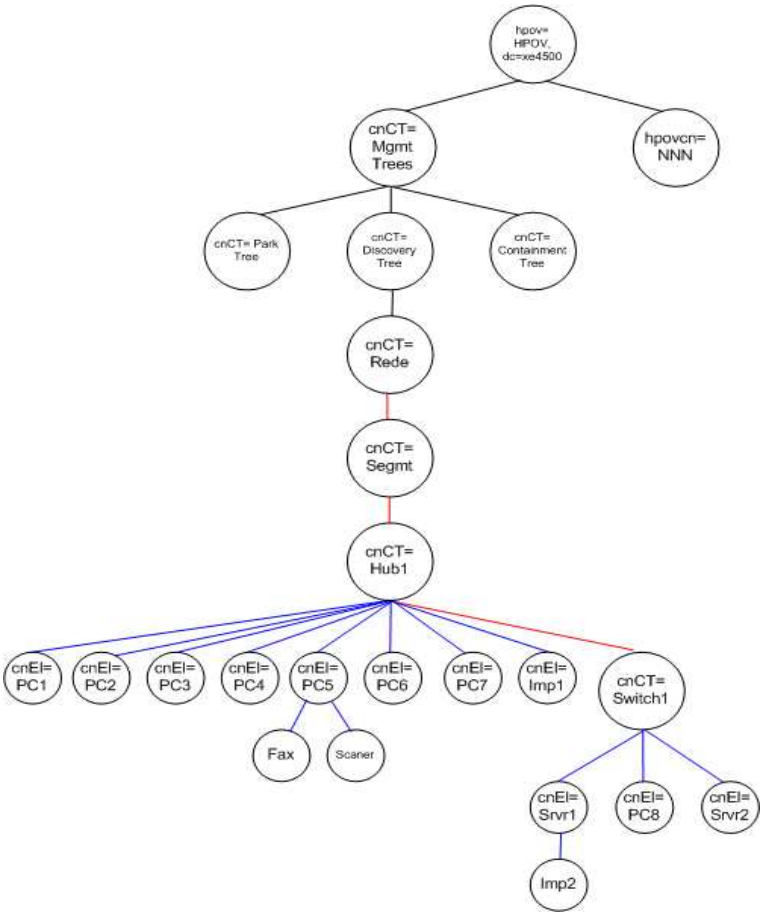

Figura 3.4: Arquetipo LDAP - Cenário 1

Cenário 3

Neste terceiro cenário temos duas situações muito interessantes pois evidênciam o modo como se podem mapear as topologias de rede, uma vez que neste caso foi introduzido na rede mais um router. Aqui importa destacar que, embora haja mais que um router, todos os segmentos pertencem à mesma rede. Caso os routers não fossem da mesma rede, seguíamos o caso do cenário quatro (cenário seguinte).

A diferença da topologia da árvore da Figura 3.8 para a topologia da

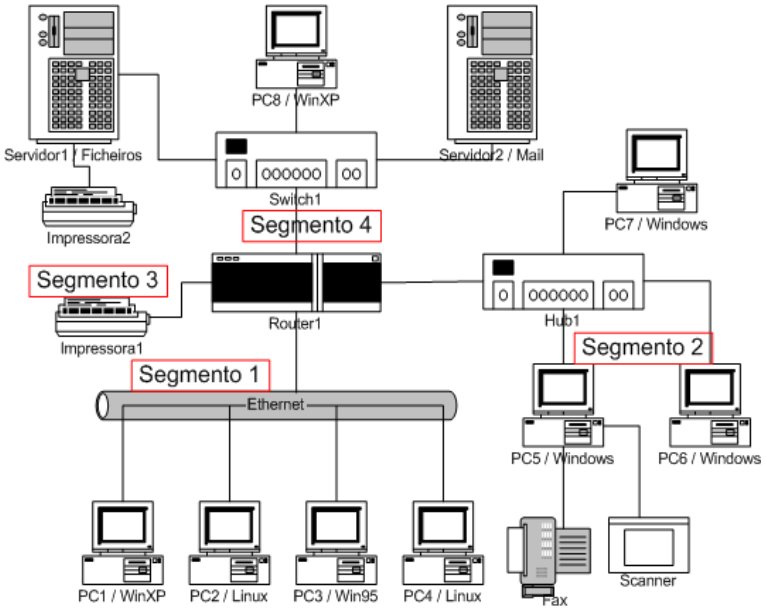

Figura 3.5: Arquetipo Rede - Cenário 2

árvore da Figura 3.9 tem a ver com o segmento pelo qual se começa a fazer a descoberta. No primeiro caso a descoberta é feita a partir do "Router1", enquanto que no segundo caso, a descoberta é feita a partir do "Router2", sendo, portanto, bem visível a diferença da disposição dos equipamentos da rede na árvore LDAP.

Outra das características visíveis na árvore de LDAP é o seu balanceamento. Se iniciarmos a fazer a descoberta dos equipamentos pelo router que estiver mais no interior ou com mais segmentos, a árvore não é tão profunda, tornando-a, assim, mais larga o que facilita bastante as pesquisas de informação.

Como se pode constatar, são notórias as diferenças entre as duas figuras. A visualização das Figuras 3.8 e 3.9, mostram que a árvore seria cada vez mais profunda se houvessem outros routers ligados ao "Router1" ou nos concentradores (equipamentos que no cenário estão conectados ao "Router1").

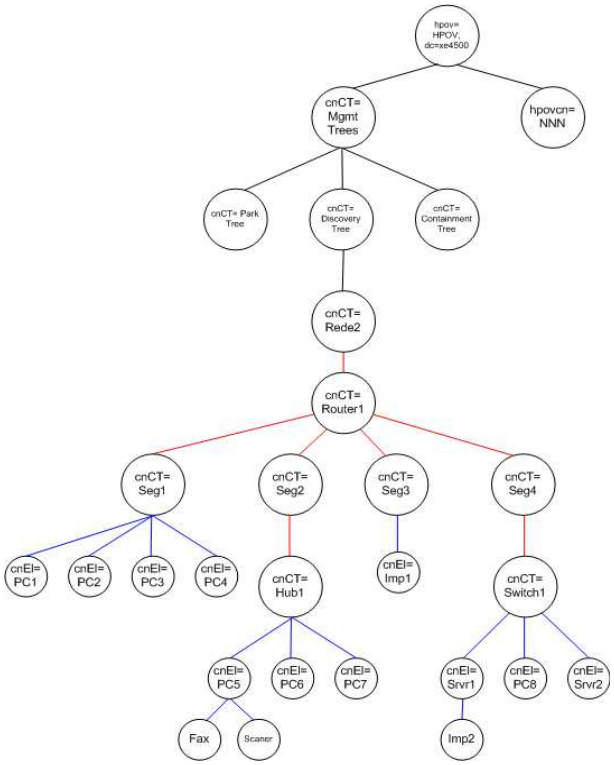

Figura 3.6: Arquetipo LDAP - Cenário 2

Cenário 4

Neste último cenário, não só tentamos abstrair-nos da parte de mapeamento dos equipamentos de rede, uma vez que essa função já foi demonstrada nas secções anteriores, mas tanbém concentraremos a nossa atenção no caso de haver várias redes, ou seja, a forma como serão mapeadas essas redes na árvore LDAP, sendo essas redes interconectadas pelo mesmo router.

Ilustrado na Figura 3.10, o router que interliga todas as redes envolventes é comum na árvore LDAP, isto é, o router aparece em todos os ramos da árvore de DT[10].

[10]Discovery Tree == Árvore de Descoberta

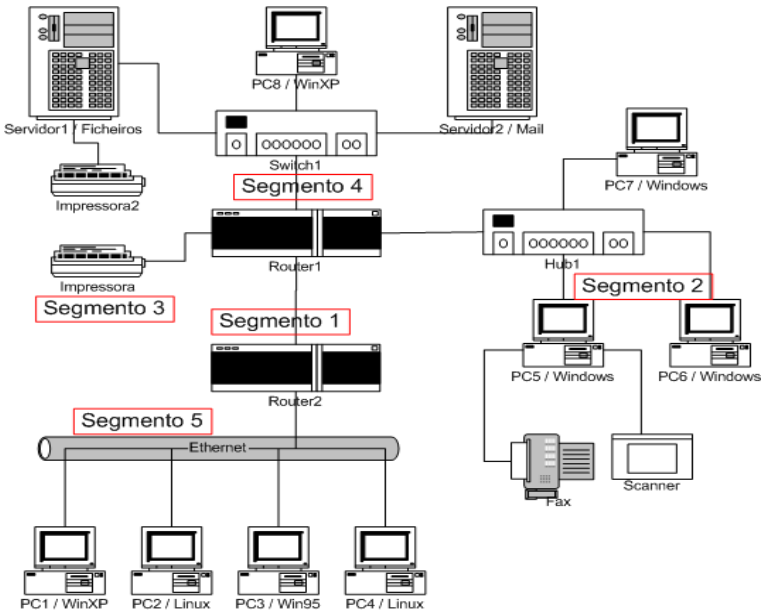

Figura 3.7: Arquetipo Rede - Cenário 3

Acrescentou-se no modelo da Figura 3.11 uma classe LDAP, que contém um apontador para a localização das outras redes. Esse apontador é um URL LDAP para a localização da definição da topologia da rede em qualquer servidor. Essas redes podem residir no mesmo servidor LDAP ou noutro servidor LDAP que integre o mapeamento dessa rede.

3.6.3 Proposta em LDAP para DT

Como se pode verificar pela Figura E.2 incluida em Apêndice, dá para ter uma visão de como fica organizada a informação relativa a "Discovery Tree". Assim desta forma fica mapeada na árvore LDAP todos os equipamentos existentes numa rede e define bem as posições e a quem está ligado fisicamente, contendo desta forma informação definida nos esquemas LDAP. Se hover algum problema em algum equipamento, pode-se resolver esse mesmo problema sabendo as suas ligaçoes, e o que isso implica para a sua resolução.

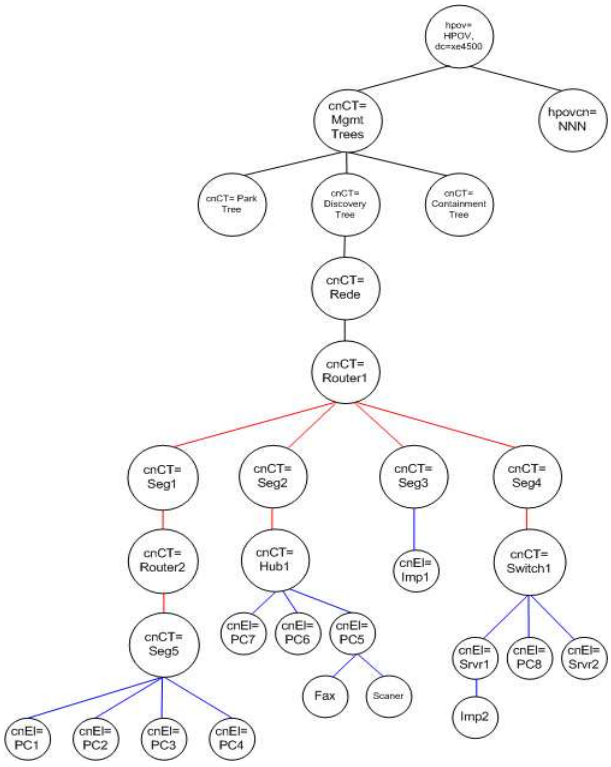

Figura 3.8: Arquetipo LDAP - Cenário 3

3.7 Árvore de Parque Informático

3.7.1 Objectivo

A Internet tem vindo a desenvolver-se cada vez mais, tornando-se uma rede mais complexa de gerir. Urge então a necessidade de desenvolver um modelo "Park Tree"[11]. A informação acerca dos nodos ou equipamentos ligados à rede é guardada numa base de dados especial, chamada de serviço de directório LDAP. Um serviço de directório fisicamente distribuído, constituindo um repositório lógicamente centralizada de dados que alteram muito pouco ou quase nunca, é usado para gerir o ambiente computacional.

O modelo apresentado não é único, existindo já alguma aplicação desta funcionalidade, por exemplo no AD da Microsoft. Este modelo tende a ser mais profundo na quantidade de informação a ser gerida, buscando por isso muita da informação aos modelos CIM e DEN para modelar o PT.

[11]Park Tree == PT == o mesmo que Árvore Parque Informático

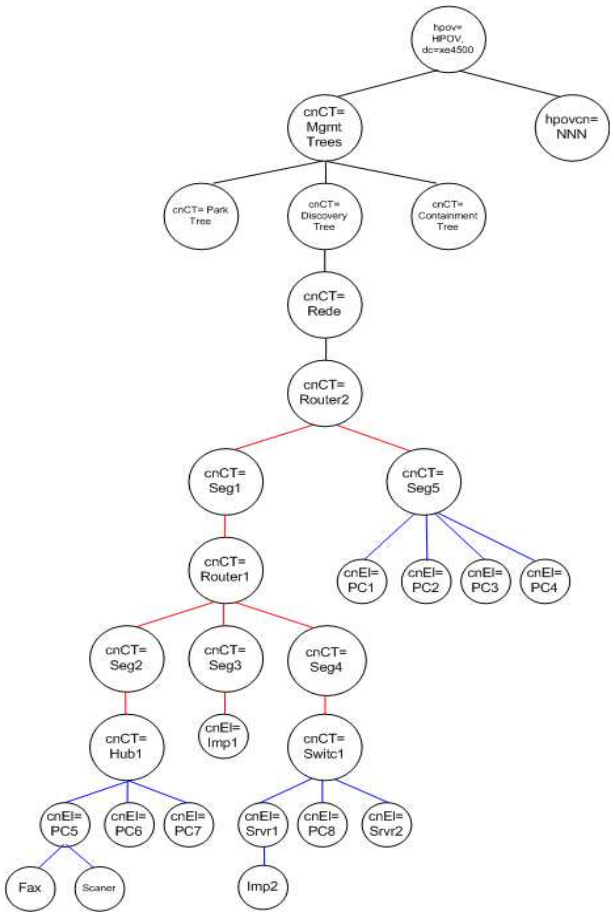

Figura 3.9: Arquetipo LDAP - Cenário 3

O PT é um modelo vantajoso para as aplicação de gestão de redes, assim como para os gestores de redes porque disponibiliza, de forma simples e clara, toda a informação dos equipamentos existentes na rede. A informação das caraterísticas físicas sobre cada equipamento pode ser obtida através do LDAP, sem ter de o inquirir, estando o equipamento on-line ou off-line.

Na secção seguinte demostra-se, com a ajuda da definição de modelos abstractos, o modo como podemos implementar em LDAP uma árvore com a informação relativa a todos os equipamentos de uma rede.

3.7.2 Arquitectura PT

Este mapeamento de informação das carecterísticas acerca dos equipamentos existentes numa rede traz muitas vantagens :

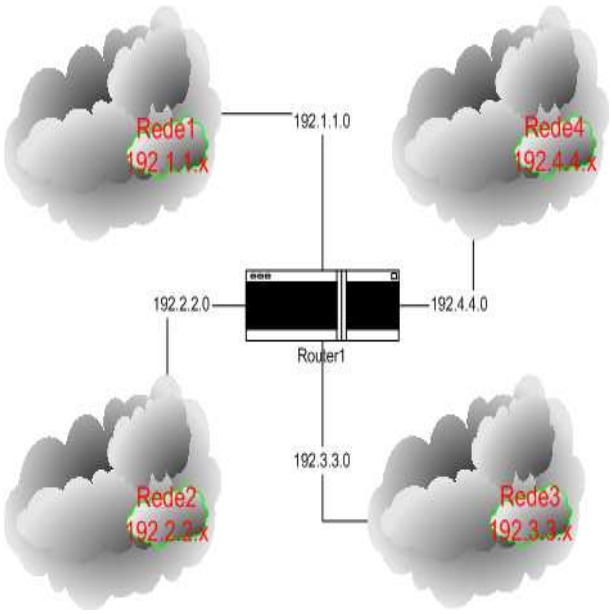

Figura 3.10: Arquetipo Rede - Cenário 4

- manter actualizada e centralizada toda a informação relativa a qualquer equipamento que faça parte da rede, ficando, por isso, o inventário do parque de máquinas actualizado;

- saber a qualquer momento as características de qualquer equipamento de rede sem ter ligação remota (por telnet, por exemplo) para tentar saber as características;

- detectar situações de resolução de problemas de conectividade, onde a informação está disponível off-line, sem ter necessidade de inquirir o equipamento quando este está do outro lado do problema;

Nas figuras desta secção, apresentam-se duas possíveis árvores DIT[12].Na figura 3.12 os equipamentos encontrados são inseridos no contentor LDAP do seu fabricante, sendo, portanto diferente da Figura 3.13 na qual todos os equipamentos estão no mesmo contentor LDAP, usando da capacidade de organização do LDAP por contentores.

Pela análise visual das árvores DIT, nas duas figuras, a seguir explícitadas, verifica-se que ambas podem ser aplicáveis em qualquer caso para melhoria de gestão da informação e da gestão de redes.

[12]DIT == Directory Information Tree

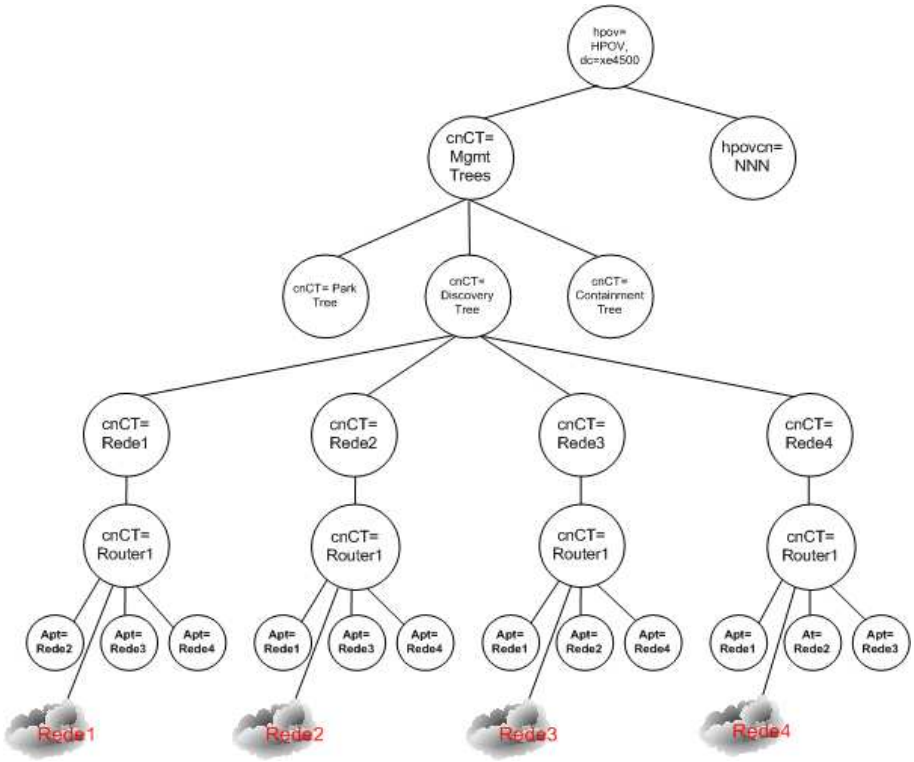

Figura 3.11: Arquetipo LDAP - Cenário 4

Existem vantagens e desvantagens do modelo da Figura 3.12 em relação ao modelo da Figura 3.13, que são:

As vantagens :

- melhor organização dos equipamentos pelo seu fabricante;
- melhor controlo de inventário dos equipamentos de um determinado fabricante;

As desvantagens :

- é necessário criar mais um contentor na árvore LDAP para guardar informação do fabricante do equipamento;
- surgem algumas dificuldades na pesquisa da informação de um determinado equipamento;

Face ao dilema da escolha da árvore de DIT para guardar a informação das características dos equipamentos encontrados em cada rede, é necessá-

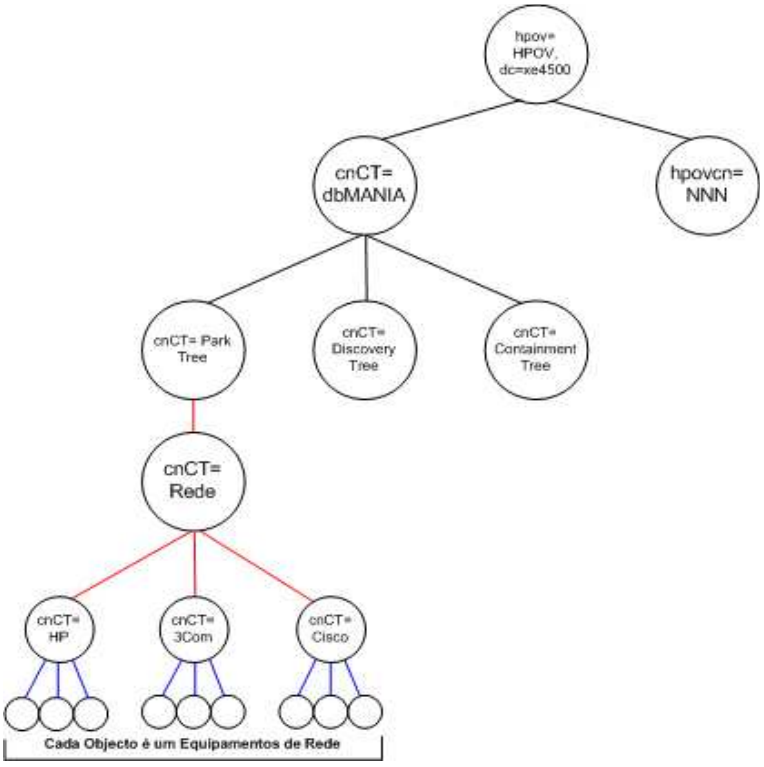

Figura 3.12: Estrutura Funcional do Park Tree com Contentores

ria muita ponderação a fim de evitar demoras nas pesquisas da árvore de directório.

Tal não significa que é obrigatório seguir estes modelos, ficando ao critério do fabricante da aplicação de gestão de redes analisar e tentar verificar se é benéfico seguir um destes modelos.

3.7.3 Proposta em LDAP para PT

Como se pode verificar pela Figura E.3 incluida em Apêndice, os equipamentos ficam organizados pelo seu fabricante. Assim desta forma a árvore fica simples e define bem as características desse mesmo equipamento, sendo bastante útil para fazer um inventário ou um upgrade ou mesmo uma substituiçao no sistema que se pretende, sem ter a necessidade de fazer remotamente a ligação ao sistema. Esta solução tem algumas vantagens, por exemplo, quando o sistema está "off-line" e necessitamos de saber todas as caracteristicas para a sua substituiçao.

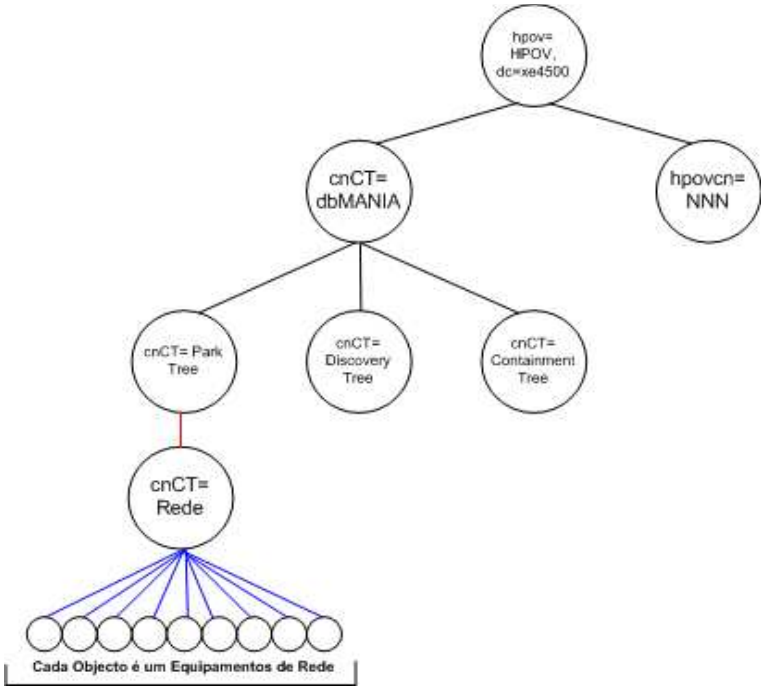

Figura 3.13: Estrutura Funcional do Park Tree sem Contentores

Capítulo 4

Conclusão

4.1 Integração Global das Redes

Actualmente o mundo da gestão de redes e de sistemas complexos propõem e necessitam de soluções funcionais standards ou soluções proprietárias.

As soluções standards apoiam-se na utilização combinada de um modelo de informação SMI ou GDMO, de um paradigma organizacional do tipo Agente/Manager e de um protocolo suportando as interacções e as trocas de informações de gestão entre as entidades (como por exemplo o SNMP, CMIP e o DMI). Cada um destes standards aplica-se a gestão de um ambiente tecnológico especifico e não a vários (o **SNMP** para as redes informáticas, o **CMIP** para as redes de telecomunicações, e o **DMI** para os PCs).

Soluções proprietárias podem igualmente ser encontradas, bem como soluções visando a supervisionar os "recursos" empregues. Em suma, tais aproximações só autorizam uma simples instrumentação, só manipulam uma informação pobre, oferecendo um fraco nível de automatização e de delegação de tarefas.

É de agora em diante necessário fornecer meios de supervisão de redes e sistemas complexos que integram estas diferentes soluções a fim de dotar os administradores de redes de uma visão global dos reais recursos, gerados independentemente das suas heterógeneidades.

O objectivo de alguns trabalhos nesta área é então definir, conceber e desenvolver infra-estruturas de gestão que permitam : [9]

- a integração e interoperabilidade das soluções existentes;
- possibilidades de dimensionamento (escalabilidade) ao ambiente gerado;
- a automatização das tarefas;

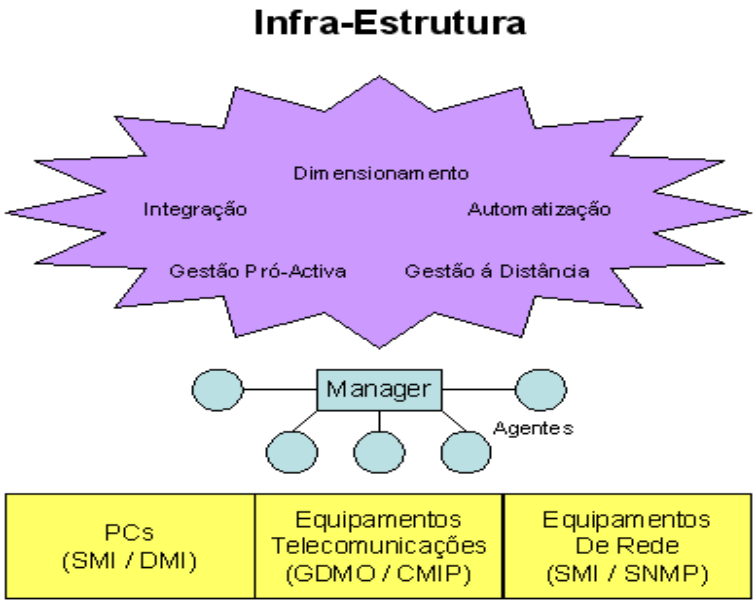

Figura 4.1: Infra-Estrutura Integrada de Gestão de Redes

- a gestão a distância;
- a gestão pró-activa;

Para que se possa atingir os cinco items da secção anterior, é necessário progressivamente realizar três acções, respectivamente : [9]

- a abertura dos sistemas de gestão;
- a supervisão dos sistemas de gestão;
- a cooperação entre sistemas de gestão;

Nos três casos, as soluções podem integrar proposições relevantes nos aspectos informativos, organizacional e protocolar a fim de beneficiar, numa solução geral, da combinação desses três aspectos complementares.

4.2 Sobre os Modelos Propostos

Para a realização deste trabalho prendeu-se fundamentalmente em desenvolver três modelos de árvores, implementadas em LDAP, como principio **basilar** para resumir e organizar toda a informação útil sobre as redes e

seus equipamentos nela inseridos, e definir algumas regras de conexão entre os componentes de redes, com o intuito de criar uma rede "inteligente". Acabou-se por elaborar os três modelos de gestão da seginte forma :

- modelo de CT (Containment Tree), para definição de regras de conexão entre os componentes/equipamentos de redes;
- modelo de DT (Discovery Tree), para mapear as topologias e os equipamentos de rede em LDAP, aqui guardam-se todas as referências das MIBs em LDAP para que as aplicações de gestão ou gestores de redes possam tratar ou "atacar" os objectos de cada MIB;
- modelo de PT (Park Tree), para obter um inventário actualizado de todo o parque de máquinas de cada rede, aqui guardam-se todas as características físicas relevantes, das máquinas existentes em cada rede;

Os modelos de CT (Containment Tree), de PT (Park Tree) e de DT (Discovery Tree) iniciados no grupo e projecto MANIA[1], que originou esta tese de Mestrado, tem como intenção promover e divulgar a criação de modelos standards e consistentes para gestão de redes com equipamentos e serviços heterogéneos, usando para isso o serviço de directório LDAP.

Estes modelos constituem uma ajuda para vários fabricantes de software de gestão de redes, tentem seguir uma normalização a fim de que outras aplicações possam utilizar essa informação de forma transparente e sem grandes dificuldades de integração e de apresentação de soluções na gestão de redes. Tal procedimento ajudará a formar e consolidar o futuro desenvolvimento de aplicações de gestão de redes.

4.3 Futuros trabalhos

É de referir que algum trabalho com o uso do LDAP está a ser elaborado nesta área de gestão de redes informáticas, todavia muito mais pode ser feito, aproveitando as capacidades do serviço de directórios, usando por exemplo, na gestão de equipamentos na área das telecomunicaçoes, ou seja, na integraçao de todos os equipamentos de redes e informáticas e equipamentos de telecomunicações, tornando como um todo.

Os trabalhos futuros a desenvolver nesta área será melhorar a organização dos atributos das classes LDAP, adicionando-lhe novos atributos á medida que vão surgindo novos equipamentos ou consolidando os existentes, contribuindo desta forma para a formalização da aquitectura em termos da sua implementação em ambiente real e não academico.

[1]MANIA == Managing Awareness in Networks through Intelligent Agents

Bibliografia

[1] www.sei.cmu.edu, *Network Management - Software Tecnology Roadmap*, February 12, 2004.

[2] www.sei.cmu.edu,*SNMP - Software Tecnology Roadmap*, February 12, 2004.

[3] www.sei.cmu.edu, *CMIP - Software Tecnology Roadmap*, February 12, 2004.

[4] www.cellsoft.de, *CMIP/CMIS - Object Oriented Network Management*, February 2004.

[5] Yu, Xinzhong, *Directory Enabled Networks*, December 3, 1998.

[6] www.cisco.com, *DEN - Frequentled Asked Questions*, March 2004.

[7] DMTF, *The Growing Importance of Management Standards*, September 2003.

[8] DMI, *The Growing Importance of Management Standards*, September 2003.

[9] Sibilla, Michelle , *IRIT Toulouse*, November 2004.

[10] Fernique, Pierre, *Un Modèle pour l'Administration des Dépendences entre Services Applicatifs des Réseaux Informatiques Hétérogènes*, Septembre 1994.

[11] Oliveira, Raul Teixeira, *Gestion des Réseaux Connaissance des Besoins: Utilisation des Agents Logiciel*, Janvier 1998.

[12] Arkills, Brian, *How LDAP Works*, January 23, 2003.

[13] Arkills, Brian, *LDAP Directories Explained: An Introduction and Analysis*, February 21, 2003

[14] Malere, Luiz Ernesto, *LDAP Linux HowTo*, September 2000.

[15] Carter, Gerald, *LDAP System Administrator*, March 2003.

[16] www.redbooks.ibm.com, *Using LDAP for Directory Integration*, February 2004.

[17] www.slac.stanford.edu, *Network Monitoring Tools*, February 2004.

[18] www.microsoft.com, *Active Directory Service Interfaces*, Setembro 2004

[19] www.assure24.com, *Assure24 MIB Database*, Outubro 2004

[20] www.isode.com, *Case Study: Novis and M-Vault*, 2003

[21] www.padl.com, *NIS/LDAP Gateway*, Outubro 2005

[22] www-03.ibm.com, *Using LDAP for Naming Services*, Outubro 2005

Apêndice A

Listagem Código CT

Nesta secção faz-se a inclusão da definição de atributos e classes em LDAP para **"Containment Tree"**, usadas neste trabalho de investigação. Serve esta informação para futuras aplicações deste modelo em novos trabalhos para melhoria da funcionalidade deste modelo um pouco simples.

```
#maniaDB.schema
#Schema for MANIA's DataBase Network Management Framework
#OIDs are owned by the MANIA Research Group
#MANIA IANA's OID Number : 19248
#Author : Jorge Coutinho<mrs01017@fe.up.pt>
#Created: CopyRight (c), May 2004
#Updated: May 2004, June 2004, July 2004, September 2004
#Version: v1.0, v1.5, v2.0, v2.5 #

#MANIA SYNTAXes

#1.3.6.1.4.1.19248 ---> MANIA Root
objectIdentifier oid_MANIA 1.3.6.1.4.1.19248

#1.3.6.1.4.1.19248.10---> MANIA DATABASE Root for Elements/Equipment
objectIdentifier oid_rootDB 1.3.6.1.4.1.19248.10

#Definition of the (branch) MANIA's Class DataBase
#1.3.6.1.4.1.19248.10.1.x ---> OID for Atributes in Discovery Tree
#1.3.6.1.4.1.19248.10.2.x ---> OID for Classe in Discovery Tree

objectIdentifier oid_ctATTR oid_rootDB:1
objectIdentifier oid_CT oid_rootDB:2
```

```
#-#-# Definition of common DB Network Management

attributetype ( oid_ATTR:1
                NAME ('cnCT' 'cnEL' 'commonNameCT' 'commonNameEL')
                DESC 'The container common name'
                EQUALITY caseIgnoreIA5Match
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{20}
                SINGLE-VALUE )

attributetype ( oid_ATTR:2
                NAME ('descCT' 'description' 'descEL' )
                DESC 'The description of the CT or of Element'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.36
                SINGLE-VALUE )

attributetype ( oid_ATTR:3
                NAME ('typeEL' 'typeEQUIP' )
                DESC 'The type of the CT or of Element'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.36
                SINGLE-VALUE )

# # # # # # BEGIN CLASSES DEFINITIONS # # # # # #

# Definition of the Class CT, DT and PT objectClass ( oid_CT:1
            NAME 'ctCT'
            DESC 'Class Definition for CT, DT and PT'
            SUP top STRUCTURAL
            MUST ( cnCT )
            MAY ( descCT )

objectClass ( oid_CT:2
            NAME 'ctEL'
            DESC 'Class Definition for ELEMENTS'
            SUP top STRUCTURAL
            MUST ( cnEL )
            MAY ( descEL $ typeEL )

# # # # # # END CLASSES DEFINITIONS # # # # # #
```

Apêndice B

Listagem Código DT

Nesta secção faz-se a inclusão da definição de atributos e classes em LDAP para **"Discovery Tree"**, usadas neste trabalho de investigação. Serve esta informação para futuras aplicações deste modelo em novos trabalhos para melhoria da funcionalidade deste modelo um pouco simples.

```
#maniaDB.schema
#Schema for MANIA's DataBase Network Management Framework
#OIDs are owned by the MANIA Research Group
#MANIA IANA's OID Number : 19248
#Author : Jorge Coutinho<mrs01017@fe.up.pt>
#Created: CopyRight (c), May 2004
#Updated: May 2004, June 2004, July 2004, September 2004
#Version: v1.0, v1.5, v2.0, v2.5 #

#MANIA SYNTAXes

#1.3.6.1.4.1.19248 ---> MANIA Root
objectIdentifier oid_MANIA 1.3.6.1.4.1.19248

#1.3.6.1.4.1.19248.10---> MANIA DATABASE Root for Elements/Equipment
objectIdentifier oid_rootDB 1.3.6.1.4.1.19248.10

#Definition of the (branch) MANIA's Class DataBase
#1.3.6.1.4.1.19248.10.3.x ---> OID for Atributes in Discovery Tree
#1.3.6.1.4.1.19248.10.4.x ---> OID for Classe in Discovery Tree

objectIdentifier oid_dtATTR oid_rootDB:3
objectIdentifier oid_DT oid_rootDB:4
```

```
#-#-# Definition of common DB Network Management

attributetype (oid_dtATTR:1
                NAME ('cnCT' 'commonNameCT' 'cnEL' 'commonNameEL' )
                DESC 'The container common name'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{20}
                SINGLE-VALUE )

attributetype ( oid_dtATTR:2
                NAME ('descCT' 'descEL' 'description')
                DESC 'The description of the CT'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
                SINGLE-VALUE )

attributetype ( oid_dtATTR:3
                NAME 'addrMAC'
                DESC 'The MAC Address of the Element'
                EQUALITY caseIgnoreIA5Match
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{20}
                MULTI-VALUE )

attributetype ( oid_dtATTR:4
                NAME 'addrIP'
                DESC 'The IP Address of the Element'
                EQUALITY caseIgnoreIA5Match
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{20}
                MULTI-VALUE )

attributetype ( oid_dtATTR:5
                NAME ('addrNET' 'addrSEG')
                DESC 'The Address of the NET or SEGMENT'
                EQUALITY caseIgnoreIA5Match
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{20}
                SINGLE-VALUE )

attributetype ( oid_dtATTR:6
```

```
                NAME ('maskNET' 'maskSEG')
                DESC 'The mask of the NET or SEGMENT'
                EQUALITY caseIgnoreIA5Match
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{20}
                SINGLE-VALUE )

attributetype ( oid_dtATTR:7
                NAME ('nameNET' 'nameSEG')
                DESC 'The name of the NET or SEGMENT'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{50}
                SINGLE-VALUE )

attributetype ( oid_dtATTR:8
                NAME ('numberElements' 'numberSEG')
                DESC 'The number of the SEGMENT or Elements'
                EQUALITY numericStringMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
                SINGLE-VALUE )

attributetype ( oid_dtATTR:9
                NAME ('ifSUP' 'interfaceSUP')
                DESC 'The interface of their SUPERIOR'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{20}
                SINGLE-VALUE )

attributetype ( oid_dtATTR:10
                NAME 'mibOID'
                DESC 'The all OID numbers that Equipment has'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{20}
                SINGLE-VALUE )


#-#-#-#-#-# BEGIN CLASSES DEFINITIONS #-#-#-#-#-#

#-# Definition of the Classes for DT
```

```
objectClass ( oid_DT:1
            NAME 'dtREDE'
            DESC 'Class Definition for REDE'
            SUP top STRUCTURAL
            MUST ( cnCT )
            MAY ( addrNET $ maskNET $ description $ nameNET $ numberElements )

objectClass ( oid_DT:2
            NAME 'dtSEG'
            DESC 'Class Definition for SEGMENT'
            SUP top STRUCTURAL
            MUST ( cnCT  )
            MAY ( addrSEG $ maskSEG $ description $ nameSEG $ numberElements )


objectClass ( oid_DT:3
            NAME 'dtROUTER'
            DESC 'Class Definition for ROUTERS'
            SUP top STRUCTURAL
            MUST ( cnEL )
            MAY ( description $ addrIP $ addrMAC $ mibOID $ numberSEG )

objectClass ( oid_DT:4
            NAME 'dtAPTNET'
            DESC 'Class Definition for Pointer to ohter NETWORK'
            SUP top STRUCTURAL
            MUST ( cnCT )
            MAY ( aptNET $ description $ interfaceSUP )

objectClass ( oid_DT:5
            NAME 'dtPC'
            DESC 'Class Definition for PCs'
            SUP top STRUCTURAL
            MUST ( cnEL )
            MAY ( description $ addrIP $ addrMAC $ mibOID $ interfaceSUP )

objectClass ( oid_DT:6
            NAME 'dtSERVER'
            DESC 'Class Definition for SERVERS'
            SUP top STRUCTURAL
            MUST ( cnEL )
            MAY ( description $ addrIP $ addrMAC $ mibOID $ interfaceSUP )
```

```
objectClass ( oid_DT:7
            NAME 'dtHUB'
            DESC 'Class Definition for HUBs'
            SUP top STRUCTURAL
            MUST ( cnEL )
            MAY ( description $ addrIP $ addrMAC $ mibOID $ interfaceSUP )

objectClass ( oid_DT:8
            NAME 'dtSWITCH'
            DESC 'Class Definition for SWITCHs'
            SUP top STRUCTURAL
            MUST ( cnEL )
            MAY ( description $ addrIP $ addrMAC $ mibOID $ interfaceSUP )

objectClass ( oid_DT:9
            NAME 'dtEL'
            DESC 'Class Definition for FAXs, PRINTERs, SCANERs, etc'
            SUP top STRUCTURAL
            MUST ( cnEL )
            MAY ( description $ interfaceSUP )

# # # # # # END CLASSES DEFINITIONS # # # # # #
```

Apêndice C

Listagem Código PT

Nesta secção faz-se a inclusão da definição de atributos e classes em LDAP para **"Park Tree"**, usadas neste trabalho de investigação. Serve esta informação para futuras aplicações deste modelo em novos trabalhos para melhoria da funcionalidade deste modelo um pouco simples.

```
#maniaDB.schema
#Schema for MANIA's DataBase Network Management Framework
#OIDs are owned by the MANIA Research Group
#MANIA IANA's OID Number : 19248
#Author : Jorge Coutinho<mrs01017@fe.up.pt>
#Created: CopyRight (c), May 2004
#Updated: May 2004, June 2004, July 2004, September 2004
#Version: v1.0, v1.5, v2.0, v2.5 #

#MANIA SYNTAXes

#1.3.6.1.4.1.19248 ---> MANIA Root
objectIdentifier oid_MANIA 1.3.6.1.4.1.19248

#1.3.6.1.4.1.19248.10---> MANIA DATABASE Root for Elements/Equipment
objectIdentifier oid_rootDB 1.3.6.1.4.1.19248.10

#Definition of the (branch) MANIA's Class DataBase
#1.3.6.1.4.1.19248.10.5.x ---> OID for Atributes in Discovery Tree
#1.3.6.1.4.1.19248.10.6.x ---> OID for Classe in Discovery Tree


objectIdentifier oid_ptATTR oid_rootDB:5
objectIdentifier oid_PT oid_rootDB:6
```

```
#-#-# Definition of common DB Network Management

attributetype ( oid_ptATTR:1
                NAME ('cnCT' 'commonNameCT' 'cnEL' 'commonNameEL' )
                DESC 'The container or Element common name'
                EQUALITY caseIgnoreIA5Match
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{20}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:2
                NAME ('descCT' 'descEL' 'description')
                DESC 'The description of the CT'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.36
                SINGLE-VALUE )

attributetype ( oid_ptATTR:10
                NAME ('manufName' 'manufacturerName')
                DESC 'The Common name for the Manufacturer'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:11
                NAME 'modelName'
                DESC 'The model name for the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:12
                NAME 'serialNumber'
                DESC 'The serial Number of the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )
```

```
attributetype ( oid_ptATTR:13
                NAME 'partNumber'
                DESC 'The part Number of the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:14
                NAME 'dateProduction'
                DESC 'The date of production of the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{20}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:15
                NAME 'vendorEquipType'
                DESC 'The vendor Equipment Type of the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:16
                NAME 'physicalPosition'
                DESC 'The physical position of the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:17
                NAME 'addrLocation'
                DESC 'The address location of the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                MULTI-VALUE )

attributetype ( oid_ptATTR:18
                NAME 'ownerName'
                DESC 'The owner name of the Equipment'
```

```
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                MULTI-VALUE )

attributetype ( oid_ptATTR:19
                NAME 'ownerContact'
                DESC 'The owner contact of the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                MULTI-VALUE )

attributetype ( oid_ptATTR:20
                NAME 'memCapacity'
                DESC 'The memory capacity of the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:21
                NAME 'diskDrive'
                DESC 'The manufacturer, model, capacity and serial number of the dis
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                MULTI-VALUE )

attributetype ( oid_ptATTR:22
                NAME ('cpuInfo' 'procInfo')
                DESC 'The physical position of the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:23
                NAME 'typeInterface'
                DESC 'The types of interface on the Equipment, LPT, COMM, ETH, ATM,
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
```

```
                MULTI-VALUE )

attributetype ( oid_ptATTR:24
                NAME ('typeOS' 'typeOperatingSystem')
                DESC 'The type of the Operating System on the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:25
                NAME ('langSupported' 'languageSupported')
                DESC 'The types of interface on the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                MULTI-VALUE )

attributetype ( oid_ptATTR:26
                NAME 'numberPorts'
                DESC 'The number of Ports on the Equipment'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:26
                NAME 'typeServices'
                DESC 'The services that Equipment offers'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                MULTI-VALUE )

attributetype ( oid_ptATTR:27
                NAME 'monitorInfo'
                DESC 'The Monitor Model and other Info'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:28
```

```
                NAME 'monitorSN'
                DESC 'The Monitor Serial Number'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:29
                NAME 'keybInfo'
                DESC 'The Keyboard Model and other Info'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:30
                NAME 'keybSN'
                DESC 'The Keyboard Serial Number'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:31
                NAME 'mbInfo'
                DESC 'The MotherBoard Model and other Info'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

attributetype ( oid_ptATTR:32
                NAME 'mbSN'
                DESC 'The MotherBoard Serial Number'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{56}
                SINGLE-VALUE )

# # # # # # BEGIN CLASSES DEFINITIONS # # # # # #

# Definition of the Class CT, DT and PT objectClass ( oid_PT:1
            NAME 'ptPC'
```

```
            DESC 'PT Class Definition for PCs'
            SUP top STRUCTURAL
            MUST ( cnCT )
            MAY ( descEL $ manufName $ modelName $ serialName $
                partNumber $ dateProduction $ vendorEquipType $
                physicalPosition $ addrLocation $ ownerName $
                ownerContact $ memCapacity $ diskDrive $
                cpuInfo $ typeInterfaces $ typeOS  $
                monitorInfo $ monitorSN $ keybInfo $ keybSN $
                mbInfo $ mbSN )
            )

objectClass ( oid_PT:2
            NAME 'ptPRINTER'
            DESC 'PT Class Definition for PRINTERs'
            SUP top STRUCTURAL
            MUST ( cnCT )
            MAY ( descEL $ manufName $ modelName $ serialName $
                partNumber $ dateProduction $ vendorEquipType $
                physicalPosition $ addrLocation $ ownerName $
                ownerContact $ memCapacity $ diskDrive $
                cpuInfo $ typeInterfaces $ typeOS $ langSupported $
                mbInfo $ mbSN )
            )

objectClass ( oid_PT:3
            NAME 'ptHUB'
            DESC 'PT Class Definition for HUBs'
            SUP top STRUCTURAL
            MUST ( cnCT )
            MAY ( descEL $ manufName $ modelName $ serialName $
                partNumber $ dateProduction $ vendorEquipType $
                physicalPosition $ addrLocation $ ownerName $
                ownerContact $ memCapacity $ diskDrive $
                cpuInfo $ typeInterfaces $ typeOS $ numberPorts $
                mbInfo $ mbSN )
            )

objectClass ( oid_PT:4
            NAME 'ptSWITCH'
            DESC 'PT Class Definition for SWITCHs'
            SUP top STRUCTURAL
            MUST ( cnCT )
```

```
            MAY ( descEL $ manufName $ modelName $ serialName $
                partNumber $ dateProduction $ vendorEquipType $
                physicalPosition $ addrLocation $ ownerName $
                ownerContact $ memCapacity $ diskDrive $
                cpuInfo $ typeInterfaces $ typeOS $ numberPorts $
                mbInfo $ mbSN )
            )

objectClass ( oid_PT:5
            NAME 'ptSERVER'
            DESC 'PT Class Definition for SERVERs'
            SUP top STRUCTURAL
            MUST ( cnCT )
            MAY ( descEL $ manufName $ modelName $ serialName $
                partNumber $ dateProduction $ vendorEquipType $
                physicalPosition $ addrLocation $ ownerName $
                ownerContact $ memCapacity $ diskDrive $
                cpuInfo $ typeInterfaces $ typeOS $ typeServices $
                monitorInfo $ monitorSN $ keybInfo $ keybSN $
                mbInfo $ mbSN )
            )


objectClass ( oid_PT:6
            NAME 'ptROUTER'
            DESC 'PT Class Definition for ROUTERs'
            SUP top STRUCTURAL
            MUST ( cnCT )
            MAY ( descEL $ manufName $ modelName $ serialName $
                partNumber $ dateProduction $ vendorEquipType $
                physicalPosition $ addrLocation $ ownerName $
                ownerContact $ memCapacity $ diskDrive $
                cpuInfo $ typeInterfaces $ typeOS $ typeServices $
                mbInfo $ mbSN )
            )

# # # # # # END CLASSES DEFINITIONS # # # # # #
```

Apêndice D

Listagem Código LDIF

Nesta secção faz-se a inclusão de uma listagem experimental no formato LDIF para a definição de atributos e classes de LDAP para **"Containment Tree"**, **"Discovery Tree"** e também para **"Park Tree"** usadas neste trabalho de investigação. Serve esta informação para futuras aplicações deste modelo em novos trabalhos para melhoria da funcionalidade deste modelo um pouco simples.

```
version: 1

# LDIF Export for: cnTree=Mgmt Tree,dc=linux,dc=pt
# Generated by phpLDAPadmin on December 19, 2004 10:29 pm
# Server: Targa LDAP (linux.pt) (192.168.1.72)
# Search Scope: sub
# Total entries: 26

# Entry 1: cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnTree=Mgmt Tree,dc=linux,dc=pt
cnTree: Mgmt Tree
descTree: Arvore que alberga CT e DT e PT
objectClass: mgmttree
objectClass: top

# Entry 2: cnTree=CTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnTree=CTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnTree: CTree
objectClass: mgmttree
objectClass: top
descTree: Containment Tree
descTree: Arvore de Conteudo
```

```
# Entry 3: cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnTree: DTree
objectClass: mgmttree
objectClass: top
descTree: Discovery Tree

# Entry 4: cnTree=PTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnTree=PTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnTree: PTree
objectClass: mgmttree
objectClass: top
descTree: Park Tree

# Entry 5: cnCT=HUB,cnTree=CTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnCT=HUB,cnTree=CTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnCT: HUB
descCT: Classe para Regras de conexao com os outros equipamentos
typeEL: HUB
typeEL: PC
typeEL: SERVER
typeEL: SWITCH
typeEL: PRINTER
typeEL: ROUTER
objectClass: ctct
objectClass: top

# Entry 6: cnCT=SWITCH,cnTree=CTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnCT=SWITCH,cnTree=CTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnCT: SWITCH
descCT: Classe para Regras de conexao com os outros equipamentos
objectClass: ctct
objectClass: top
typeEL: HUB
typeEL: PC
typeEL: SERVER
typeEL: SWITCH
typeEL: PRINTER
typeEL: ROUTER

# Entry 7: cnCT=ROUTER,cnTree=CTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnCT=ROUTER,cnTree=CTree,cnTree=Mgmt Tree,dc=linux,dc=pt
```

```
cnCT: ROUTER
descCT: Classe para Regras de conexao com os outros equipamentos
objectClass: ctct
objectClass: top
typeEL: HUB
typeEL: PC
typeEL: SERVER
typeEL: SWITCH
typeEL: PRINTER
typeEL: ROUTER

# Entry 8: cnPT=3Com Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnPT=3Com Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnPT: 3Com Systems
descPT: Manufacturer of Equipments 3Com
snmpOID: 1.3.6.1.4.1.43
objectClass: ptpt
objectClass: top

# Entry 9: cnPT=HP Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnPT=HP Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnPT: HP Systems
descPT: Manufacturer of Equipments 3Com
objectClass: ptpt
objectClass: top
snmpOID: 1.3.6.1.4.1.11

# Entry 10: cnPT=Cisco Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=linux,dc=...
dn: cnPT=Cisco Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnPT: Cisco Systems
descPT: Manufacturer of Equipments 3Com
objectClass: ptpt
objectClass: top
snmpOID: 1.3.6.1.4.1.9

# Entry 11: cnEL=Hub_3Com_1,cnPT=3Com Systems,cnTree=PTree,cnTree=Mgmt Tr...
dn: cnEL=Hub_3Com_1,cnPT=3Com Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=linux
 ,dc=pt
cnCT: Hub_3Com_1
addrLocation: Sala Desenvolvimento
cpuInfo: Intel
serialNumber: serial-3com-number
partNumber: part-3com-number
```

```
objectClass: pthub
objectClass: top

# Entry 12: cnEL=Router_Cisco_1,cnPT=Cisco Systems,cnTree=PTree,cnTree=Mg...
dn: cnEL=Router_Cisco_1,cnPT=Cisco Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=
 linux,dc=pt
cnCT: Router_Cisco_1
addrLocation: Empresa XPTO
descCT: Router da Cisco
physicalPosition: Sala Servidores
serialNumber: serial-cisco-number
ownerContact: por telefone
modelName: X20-MZ
ownerName: Jorge Coutinho
typeInterface: ADSL
typeInterface: Ethernet
typeInterface: RDIS
objectClass: ptrouter
objectClass: top

# Entry 13: cnEL=PC_HP_1,cnPT=HP Systems,cnTree=PTree,cnTree=Mgmt Tree,dc...
dn: cnEL=PC_HP_1,cnPT=HP Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=linux,dc=p
 t
cnCT: PC_HP_1
memCapacity: 512
cpuInfo: Alpha
dateProduction: 03Novembro1970
serialNumber: serial-hp-number
partNumber: part-hp-number
typeInterface: FastEthernet
typeInterface: GigaEthernet
typeOS: Windows XP SP2
vendorEquipType: Linux Corporation Systems
ownerContact: Jorge Coutinho
objectClass: ptpc
objectClass: top

# Entry 14: cnDT=feup.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnDT=feup.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnDT: feup.pt
maskNET: 255.255.255.0
nameNET: Rede_Feup_Portugal
objectClass: dtrede
```

```
objectClass: top
addrNET: 192.168.100.0

# Entry 15: cnDT=uminho.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnDT=uminho.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnDT: uminho.pt
maskNET: 255.255.255.0
objectClass: dtrede
objectClass: top
addrNET: 192.168.130.0
nameNET: Rede_UMinho_Portugal

# Entry 16: cnDT=ucoimbra.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnDT=ucoimbra.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnDT: ucoimbra.pt
maskNET: 255.255.255.0
objectClass: dtrede
objectClass: top
addrNET: 192.168.110.0
nameNET: Rede_Coimbra_Portugal

# Entry 17: cnDT=umadeira.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
dn: cnDT=umadeira.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnDT: umadeira.pt
maskNET: 255.255.255.0
objectClass: dtrede
objectClass: top
addrNET: 192.168.120.0
nameNET: Rede_UMadeira_Portugal

# Entry 18: cnEL=Router_Cisco_1,cnDT=feup.pt,cnTree=DTree,cnTree=Mgmt Tre...
dn: cnEL=Router_Cisco_1,cnDT=feup.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,
 dc=pt
cnCT: Router_Cisco_1
addrIP: 192.168.192.254
addrMAC: 00.11.22.33.44.55
mibOID: 1.3.6.1.4.1.15
mibOID: 1.3.6.1.4.1.155
mibOID: 1.3.6.1.4.1.255
objectClass: dtrouter
objectClass: top
numberElements: 3
```

```
# Entry 19: cnDT=Segment_1,cnEL=Router_Cisco_1,cnDT=feup.pt,cnTree=DTree,...
dn: cnDT=Segment_1,cnEL=Router_Cisco_1,cnDT=feup.pt,cnTree=DTree,cnTree=Mgmt
  Tree,dc=linux,dc=pt
cnDT: Segment_1
addrNET: 192.168.192.0
maskNET: 255.255.255.128
nameNET: SubRede1_FEUP
description: Sub Rede do Centro Informatica FEUP
objectClass: dtseg
objectClass: top

# Entry 20: cnDT=Segment_2,cnEL=Router_Cisco_1,cnDT=feup.pt,cnTree=DTree,...
dn: cnDT=Segment_2,cnEL=Router_Cisco_1,cnDT=feup.pt,cnTree=DTree,cnTree=Mgmt
  Tree,dc=linux,dc=pt
cnDT: Segment_2
description: Sub Rede do Centro Informatica FEUP
maskNET: 255.255.255.128
nameNET: SubRede1_FEUP
objectClass: dtseg
objectClass: top
addrNET: 192.168.192.128

# Entry 21: cnEL=Hub_3Com_1,cnDT=Segment_1,cnEL=Router_Cisco_1,cnDT=feup....
dn: cnEL=Hub_3Com_1,cnDT=Segment_1,cnEL=Router_Cisco_1,cnDT=feup.pt,cnTree=D
 Tree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnCT: Hub_3Com_1
addrMAC: 00.12.23.34.45.56
mibOID: 1.2.3.4.5.6.7
mibOID: 1.2.3.4.5.6.8
mibOID: 1.2.3.4.5.6.9
ifSUP: Ethernet 0
objectClass: dthub
objectClass: top

# Entry 22: cnEL=Switch_Cisco_1,cnPT=Cisco Systems,cnTree=PTree,cnTree=Mg...
dn: cnEL=Switch_Cisco_1,cnPT=Cisco Systems,cnTree=PTree,cnTree=Mgmt Tree,dc=
 linux,dc=pt
cnCT: Switch_Cisco_1
cpuInfo: Alpha
vendorEquipType: Linux Corporate Systems
typeOS: netbuilder os v.0.16
memCapacity: 16
physicalPosition: Perto da Sala de Reunioes
```

```
objectClass: ptswitch
objectClass: top

# Entry 23: cnEL=Switch_Cisco_1,cnDT=Segment_2,cnEL=Router_Cisco_1,cnDT=f...
dn: cnEL=Switch_Cisco_1,cnDT=Segment_2,cnEL=Router_Cisco_1,cnDT=feup.pt,cnTr
 ee=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnCT: Switch_Cisco_1
addrIP: 192.168.192.253
addrMAC: 10.20.30.40.50.60
ifSUP: Ethernet 0
mibOID: 1.6.1.4.3.4.5.20
mibOID: 1.6.1.4.3.4.5.30
mibOID: 1.6.1.4.3.4.5.40
objectClass: dtswitch
objectClass: top

# Entry 24: cnEL=PC_HP_1,cnEL=Hub_3Com_1,cnDT=Segment_1,cnEL=Router_Cisco...
dn: cnEL=PC_HP_1,cnEL=Hub_3Com_1,cnDT=Segment_1,cnEL=Router_Cisco_1,cnDT=feu
 p.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnCT: PC_HP_1
addrMAC: 66.77.88.99.00.11
addrIP: 192.168.100.130
mibOID: 1.6.1.4.3.4.5.20
mibOID: 1.6.1.4.3.4.5.30
mibOID: 1.6.1.4.3.4.5.40
objectClass: dtpc
objectClass: top

# Entry 25: cnEL=Impressora_HP_1,cnEL=Hub_3Com_1,cnDT=Segment_1,cnEL=Rout...
dn: cnEL=Impressora_HP_1,cnEL=Hub_3Com_1,cnDT=Segment_1,cnEL=Router_Cisco_1,
 cnDT=feup.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=linux,dc=pt
cnCT: Impressora_HP_1
ifSUP: Ethernet 1
addrMAC: 01.02.03.04.05.06
mibOID: 1.2.3.4.5.6.8
mibOID: 1.2.3.4.5.6.9
objectClass: dtprinter
objectClass: top

# Entry 26: cnDT=uminho.pt,cnEL=Router_Cisco_1,cnDT=feup.pt,cnTree=DTree,...
dn: cnDT=uminho.pt,cnEL=Router_Cisco_1,cnDT=feup.pt,cnTree=DTree,cnTree=Mgmt
  Tree,dc=linux,dc=pt
cnDT: uminho.pt
```

```
aptNET: ldap://localhost/cnDT=uminho.pt,cnTree=DTree,cnTree=Mgmt Tree,dc=lin
 ux,dc=pt
objectClass: dtaptnet
objectClass: top

# # # # # # END LDIF DEFINITIONS # # # # # #
```

Apêndice E

Snapshots de um Browser LDAP

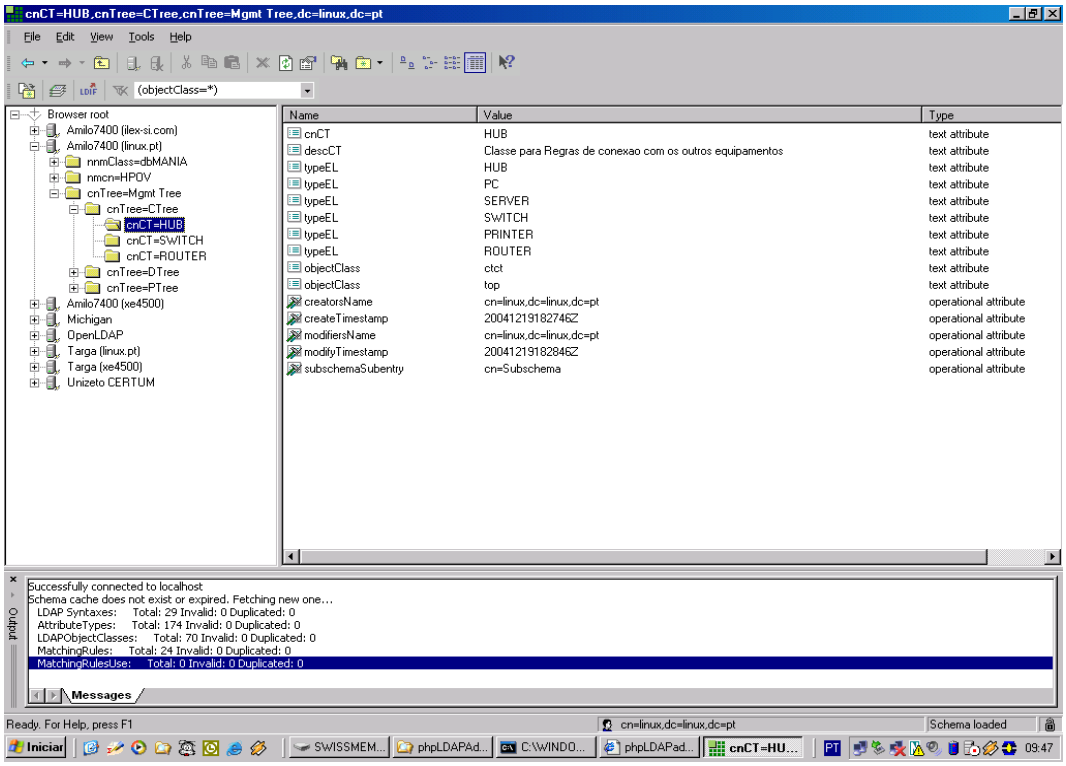

Figura E.1: Snapshot do Browser LDAP para CT

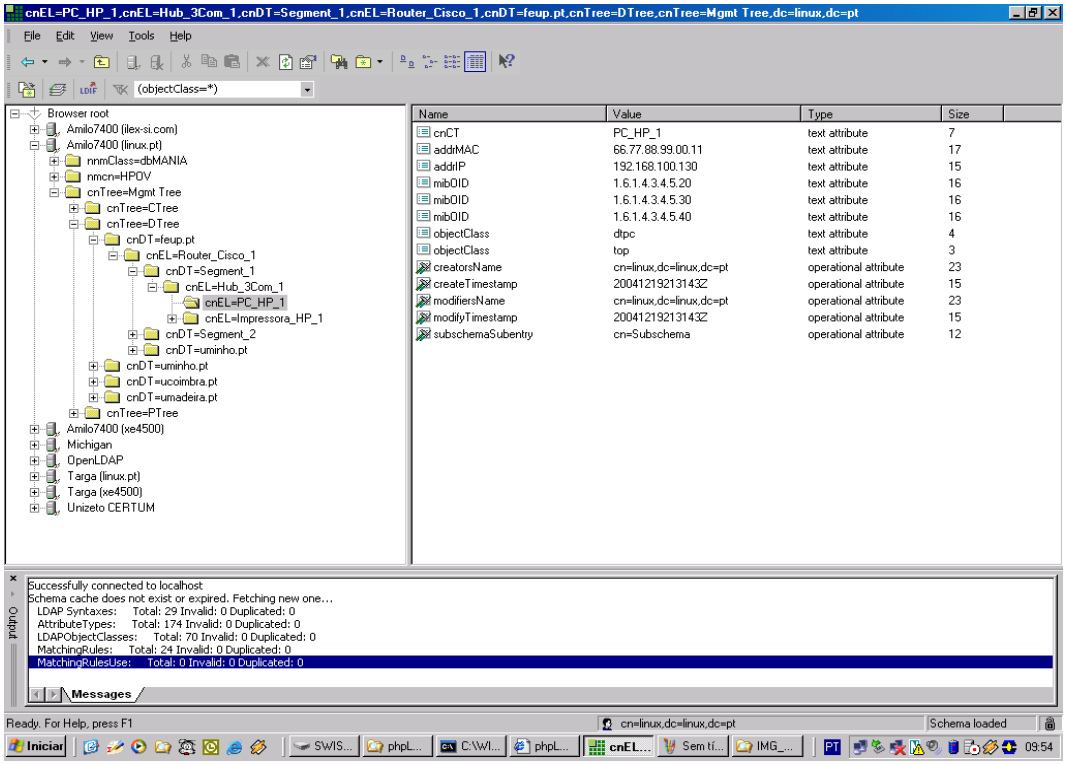

Figura E.2: Snapshot do Browser LDAP para DT

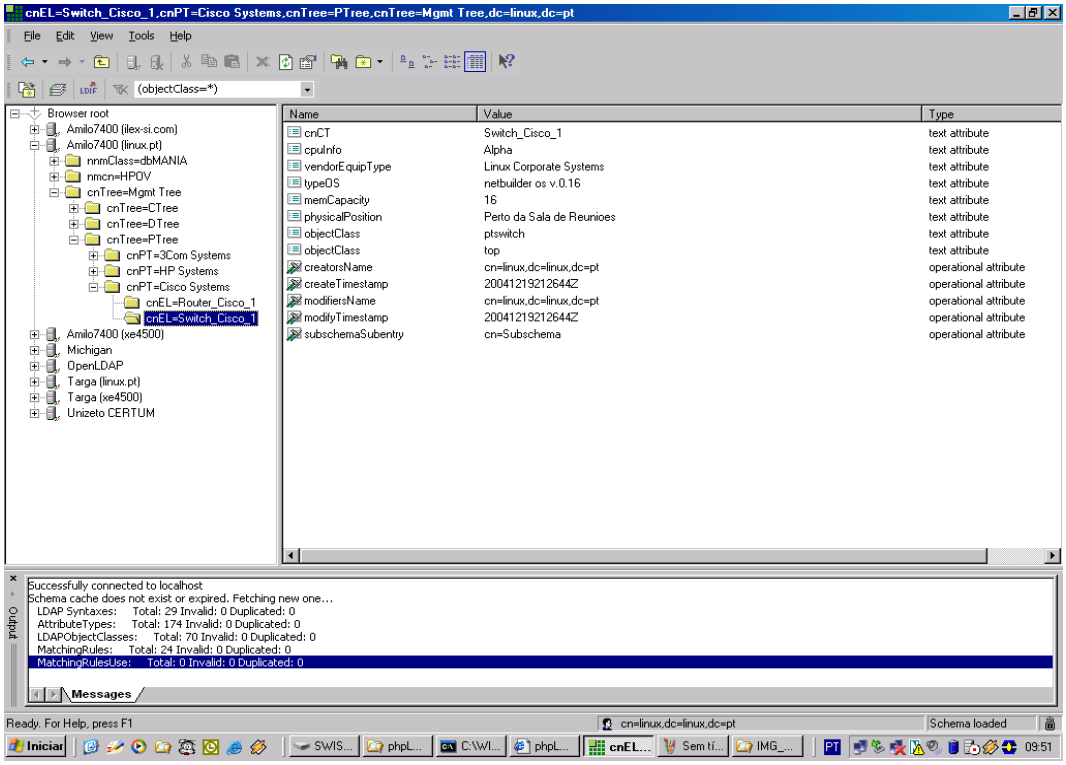

Figura E.3: Snapshot do Browser LDAP para PT