**Implantação de correio eletrônico
em ambientes heterogêneos**

Mônica M. R. Germano Travieso

Orientador: *Prof. Dr. Jan F. W. Slacks*

Dissertação apresentada ao
Instituto de Ciências Matemáticas de São Carlos,
da Universidade de São Paulo,
como parte dos requisitos para a obtenção do título
de Mestre em Ciência da Computação

São Carlos
1991

Agradecimentos

Ao meu pai, minha mãe e minha irmã por terem constituído o ambiente em que eu cresci, e pelo respeito e consideração que norteiam a nossa convivência atual. Ao meu pai, por nunca ter dado o peixe, muito menos ensinado a pescar, mas sempre estimulado a aprender a pescar. A minha mãe por sempre provar a força e potencial do ser humano, apesar de uma aparente fragilidade. A minha irmã pelo companheirismo de infância e adolescência que proporcionou referências mútuas de crescimento.

Ao Jan, nosso guru e orientador que sempre acha uma luz no fundo do túnel com bom humor e presteza.

Ao grupo de Instrumentação Eletrônica que contribui desde 1982 para minha formação, tanto na graduação, estágios e pós-graduação.

Ao departamento de Eletricidade, em especial a prof. Annie que teve especial atuação em minha ênfase em eletrônica na graduação, e nos vários cursos de pós-graduação.

Aos profs. Marcos e Regina Santana que tanto contribuíram na minha formação no curso de computação, não só em cursos, mas também em orientações e monitorias.

Ao Gomide por sempre ter concedido atenção e respeito em nossas interações, o que possibilitou esclarecimentos de muitos pontos nesse trabalho.

Ao Cláudio pela competência e disposição que tanto ajudou na decifração de vários aspectos do DECNET e PMDF.

A Agma, Caetano e Paniago que me iniciaram nas aventuras do sistema UNIX.

Ao Totó e Paniago pelas dicas e pesquisas nas redes.

Aos prof. Edson, Marcos e Regina pelas sugestões a partir do exame de qualificação.

Ao Edson por ter me recebido como estagiaria no LIE com atenção e paciência que permitiu a minha permanência e ligação ao grupo desde 1982.

Ao Lírio por tantas e tantas dicas

A Benê por ter transformado os esboços ilegíveis desse trabalho em figuras claras e bem proporcionadas.

Ao Álvaro que foi usuário pioneiro do sistema de C.E., desde as experiências com o Cirandão.

À Lia, Ivanilda, Marquinhos e Ailton por quebrarem tantos galhos sempre com disposição e bom humor.

Ao amigo Iedo por mostrar sempre que contemplação, simplicidade e espontaneidade são fundamentais.

À Adalgisa por ter me dado a chance de conviver com alguém que realizou ideais.

À Adriana pela maturidade e competência tão precoces.

Ao Gonzalo por compartilharmos todos os aspectos de nossas vidas, sempre buscando o verdadeiro caminho para um crescimento conjunto.

*Ao Gonzalo
e ao nené, que nos escolheu.*

Resumo

O presente trabalho descreve a implementação de um sistema de correio eletrônico numa rede heterogênea de computadores.

Inicialmente, faz-se uma revisão de aspectos gerais de correio eletrônico, seguida de uma discussão da interpretação de endereços e da necessidade de translação, de forma a determinar o mecanismo de transporte a ser utilizado entre os sistemas heterogêneos envolvidos.

Após uma visão geral dos protocolos utilizados, é apresentada uma descrição detalhada das configurações e interconexões usadas para implementar o sistema de correio.

Abstract

The present work describes the implementation of an E-mail system on a heterogeneous computer network.

First a review of generic aspects of E-mail systems is presented followed by a discussion of the addressing interpretation and the necessary translation in order to determine the transport mechanism to be used between the heterogeneous systems involved.

After an overview about utilized protocols, a detailed description of the final configurations and interconnections used to implement the E-mail system is presented.

Conteúdo

1	Introdução	7
1.1	Histórico e Motivação	7
1.2	Importância de correio eletrônico	9
1.3	Objetivo e Apresentação	10
2	Correio eletrônico e Redes Internacionais	11
2.1	Algumas Redes Proeminentes	11
2.2	Denominação de Recurso e Exemplos no IFQSC	21
3	Estruturação de um sistema de Correio Eletrônico	29
3.1	Identificação de Etapas	29
3.2	Transmissão por meio físico	33
3.3	Interface de usuário	35
3.4	Demais Etapas	39
4	Tradução de Denominações e Determinação do Mecanismo de Transporte	43
4.1	Descrição do despachador PMDF	44
4.2	Informações sobre o Despachador Sendmail	52
5	Protocolagem	73
5.1	Características do padrão TCP/IP	73
5.2	Tratamento dado pela DECNET ao problema de Protocolagem	78
5.3	Informações sobre o conjunto de programas UUCP	88
6	Descrição do sistema implementado	93
6.1	Tratamento dado ao DECNET e seu relacionamento com o PMDF	94
6.2	Adaptação do pacote DECUS-UUCP	98
6.3	Acomodação do programa UUPC e Circuito de Arbitramento	102
6.4	Tratamento dado ao conjunto de protocolos TCP/IP	108
6.5	Adaptação do despachador PMDF	119
6.6	Acomodação do despachador Sendmail	122
6.7	Exemplos do funcionamento do Sistema	126
7	Conclusão	139
A	Arquivos modificados para adaptar o PMDF ao Decus-UUCP	141

B	Trechos de programas que incorporam a lógica do circuito de arbitramento	147
C	Arquivo de Configuração do PMDF para o USPFSC	151
D	Arquivo de configuração do Sendmail para a estação de trabalho <i>floyd</i>	157
E	Arquivo de configuração do Sendmail para a estação de trabalho <i>emilia</i>	165
F	Arquivo de configuração do Sendmail para a estação de trabalho <i>ethos</i>	171
G	Glossário	179

Lista de Figuras

2.1	Sistema Heterogêneo que constitui a DARPA Internet	16
2.2	Tráfico de Pacotes na NSFnet	16
3.1	Ambiente do Sistema Implementado	30
3.2	O que está envolvido no envio de uma mensagem via C.E.	30
3.3	Software e Hardware nas várias etapas de C.E.	32
3.4	Opções descartadas	33
3.5	Equipamentos interligados com a identificação do Software e Hardware envolvidos na primeira e segunda etapas de C.E.	42
4.1	Relação entre os conjuntos de Regras	62
5.1	Utilização de máscaras	75
5.2	Utilização de MAX AREA como filtro	85
5.3	Sistema UUCP	91
6.1	Conexão entre USPFSC e FPSP sobre X.25	95
6.2	Filtro aplicado ao número de nó	97
6.3	Circuito de arbitramento	106
6.4	Registradores de controle e estado da 8250	108
6.5	Endereços TCP/IP	109
6.6	Domínios Internet	110
6.7	Uso de sinônimos pelo Sendmail	124
6.8	Correspondência do USPFSC a usp@cithex.hepnet	127
6.9	Correspondência de pcmonic a jif	128
6.10	Correspondência do jif ao pcmonic	130
6.11	Correspondência do xtjan ao uepaesc	131
6.12	Correspondência de uepaesc a ntiaa	133
6.13	Correspondência da emilia para o MVAXFT	135
6.14	Correspondência do USPFSC à ethos	136

Lista de Tabelas

2.1	Características de algumas Redes Proeminentes	15
2.2	Sintaxes de encaminhamento de correspondências entre redes	25
3.1	Diretrizes para serviços em X.400	40
4.1	Prefixos X Tipo de Canal	53
4.2	Chaves modificadoras opcionais	54
4.3	Macros Minúsculas	59
4.4	Meta-Símbolos	61
5.1	Camadas do TCP/IP	74
5.2	Classes dos números de rede Internet	75

Capítulo 1

Introdução

1.1 Histórico e Motivação

Desde a implantação de um sistema multiusuários VAX, da Digital, no IFQSC (Instituto de Física e Química de São Carlos), que o correio eletrônico é utilizado para troca de mensagens entre usuários com conta nesse computador.

A necessidade de correio eletrônico de maior alcance surgiu para viabilizar a troca de mensagens entre docentes do IFQSC e da Argentina, que participavam de um mesmo comitê de organização e programa de um congresso. As soluções possíveis eram: docentes daqui utilizarem o serviço de correio público da Argentina (Delphi), ou docentes da Argentina utilizarem o serviço de correios do Cirandão, do Brasil. A solução do IFQSC, com conta no Cirandão, assinar um contrato de conta subsidiária para acesso dos Argentinos foi na época a mais barata.

Essa opção permitiu também o contato com bolsistas no exterior, que tinham acesso a redes públicas, e a troca com eles de informações urgentes.

O contrato Interdata do IFQSC com a Embratel, destinado inicialmente a viabilizar o acesso da biblioteca do Instituto a bases internacionais, como Dialog, permitiu também o acesso a computadores do exterior (dos quais se conheciam contas) que estivessem conectados a redes públicas.

A par disso, eram adquiridas experiências em ligações microcomputador-*modem*, necessárias para as conexões acima.

A proliferação de microcomputadores, e a necessidade de troca de informações com os computadores da Digital também contribuíram para a aquisição de experiências.

Posteriormente, a conexão X.25 (através do pacote PSI) do VAX com a Rempac permitiu que mais usuários tivessem os acessos acima, pois se abriu a possibilidade a todos os usuários do VAX, e a ligação X.25 contratada possuía 4 canais lógicos.

O pacote PSI tornava possível o acesso a computadores VAX das redes públicas via correio eletrônico, através da sintaxe:

```
MAIL> send  
To: psi%endereco-X.25"usuario"
```

onde *endereco-X.25* é o número de endereço do computador que deve receber a correspondência. Isso foi um grande avanço, pois não exigia que se tivesse conta num computador do exterior.

A conexão da FAPESP ao Fermilab, bem como a ligação através da Renpac, permitiu o acesso do IFQSC a nós BITNET (via o pacote JNET existente na Fapesp) e a nós HEPNET (DECNET). Porém, a conexão não era nada transparente. Para acesso via correio eletrônico era necessário:

```
MAIL> send
To: psi%endereco-X.25-fapesp::no::usuario
```

A conexão DECNET, em cima de X.25, do VAX do IFQSC com a Fapesp tornou o processo mais transparente. Poderia-se acessar nós Internet com a sintaxe de endereço um pouco mais simplificada:

```
MAIL> send
To: lbl::"endereco-internet"
```

ou

```
MAIL> send
To: csa::"endereco-internet"
```

Usava-se com isso uma conexão DECNET com o nó LBL (ou CSA) da HEPNET, que possui um despachador que interpreta o endereço Internet e envia a mensagem para as várias redes científicas das quais é entroncamento.

A compra posterior do pacote despachador PMDF pela Fapesp facilitou ainda mais o processo, pois poderia-se utilizar a Fapesp, ao invés do LBL (ou CSA). Ou seja, era necessário ser assegurada apenas a conexão IFQSC-Fapesp para ser possível a entrega da mensagem, ao invés de depender das várias conexões até o LBL (ou CSA). No entanto, ainda existia o problema de que se a conexão com a Fapesp não fosse bem sucedida, o usuário não conseguiria enviar a mensagem, e teria que tentar novamente em outra hora.

A compra do PMDF pelo IFQSC permitiu que as mensagens pudessem ser enfileiradas aqui, e enviadas pelo próprio PMDF num momento em que existisse conexão. A sintaxe para enviar se tornou então:

```
MAIL> send
To: in%"endereco-Internet"
```

Enquanto isso, a cultura de redes já se impregnava no IFQSC: acesso com TCP/IP a bases de domínio público via computadores do exterior nos quais se tinha conta; conhecimento da existência de máquinas servidoras para armazenagem de software que atendem solicitação via correio eletrônico; troca de mensagens mais constante com o exterior; etc.

O aparecimento no mercado de placas que permitem a conexão de microcomputadores via Ethernet aumentou as possibilidades de ligações internas: serial ou Ethernet.

Juntamente com a chegada de novas estações de trabalho de vários fabricantes e a necessidade de transferência de dados entre os vários equipamentos, esses fatos indicaram a perspectiva de um correio eletrônico interno acoplado ao externo, num ambiente heterogêneo de máquinas e sistemas.

1.2 Importância de correio eletrônico

A maioria dos funcionários de escritório (gerentes, administradores, engenheiros, etc.) gastam grande parte de seu tempo em atividades de comunicação de informação: coleta, criação e processamento de informação em várias formas.

Uma comunicação rápida melhora a tomada de decisões e a expedição de resultados. Além disso, o que se precisa, é de ferramentas para automatizar as ações sobre uma correspondência. Uma vez em sua escrivaninha, a informação deve ser assimilada rapidamente e uma resposta enviada. Então alguém deve passar isso para outros, telefonar para quem enviou, arquivar para futuras referências - tudo em questão de minutos.

A integração da comunicação com o processamento de escritório faz do *correio eletrônico* uma ferramenta efetiva de produtividade, principalmente se seu alcance for de âmbito local e internacional.

O valor do *correio eletrônico* trouxe surpresa aos que desenvolveram a ARPANET, que esperavam que a rede fosse principalmente usada para comunicação computador a computador afim de compartilhamento de recursos, mas ao invés disso descobriram que o principal uso era *correio eletrônico*. O *correio eletrônico* funciona para a comunidade científica como parte de uma dúzia de ferramentas (programas) que são utilizadas para um grande número de propósitos. *Correio eletrônico* pode ser usado para questionar um colega afim de obter ajuda, estabelecer visitas, prever reuniões, anunciar reuniões, solicitar informação, comunicar-se com uma secretária ou administrador, comunicar-se com agências financiadoras, e assim por diante. Além dessas atividades que fazem parte do dia-a-dia do cientista, o correio é usado para funções mais específicas da comunicação científica, tais como: reagir a artigos, esclarecer pontos com estudantes e colegas, enviar relatórios e artigos para julgamento, etc.

É útil analisar porque o *correio eletrônico* tem tido um papel firme dentro das comunidades científicas, já que cientistas possuem outras ferramentas para efetuar essas funções de comunicação, tais como: conversa face-a-face, telefone, sistemas de memorandum departamental e cartas. O *correio eletrônico* assume algumas funções do serviço postal e do sistema telefônico, mas também provê serviços que esses métodos de comunicação convencionais não possuem. O *correio eletrônico* é rápido, as mensagens são entregues geralmente em segundos ou minutos, exceto em caso de falhas da rede e/ou dos entroncamentos. Diversamente de uma chamada telefônica, a chegada de uma mensagem eletrônica não necessita interromper o receptor. A caixa de correspondência provê o armazenamento temporário de mensagens permitindo ao que envia e ao que recebe trabalharem em lugares e instantes diferentes e ainda serem capazes de comunicar-se. Se necessário, o usuário pode imprimir a mensagem. Tipicamente, um usuário lerá a correspondência uma ou duas vezes por dia, conforme sua conveniência, e processará todas as mensagens que chegaram desde a última leitura. Ele responderá a algumas mensagens a medida que as lê, poderá arquivar outras de acordo com categorias que tenha estabelecido. Quando viaja, uma pessoa ainda poderá ler e processar correspondência usando um acesso de rede.

Intuitivamente se associa *correio eletrônico* a uma rede de computadores pela qual a mensagem trafega. No entanto, existem várias redes de computadores, cada uma atendendo às necessidades de uma comunidade de usuários. As diversificadas necessidades dos usuários fazem com que essas redes apresentem tantas diferenças, que é impraticável a consideração de fundi-las numa única rede. Assim para o *correio eletrônico* ter um

alcance efetivo é necessário que a mensagem possa trafegar por diferentes redes, interconectadas de modo a permitir esse tráfego.

Conforme [LJF86], a ARPANET, BITNET, CSNET, e UUCPnet, possuem diferentes arquiteturas de rede, protocolos e comunidades de usuários. No entanto, cada uma suporta um sistema de *correio eletrônico* com formatos de mensagem mais ou menos conforme o padrão DARPA Internet. A adoção de um único formato de correspondência simplifica a implementação de entroncamentos, chamados retransmissores de correspondência, entre várias redes. Esses retransmissores de correspondência são hospedeiros especiais que são conectados a duas ou mais redes e que entendem o protocolo de cada uma delas. Uma mensagem endereçada a um hospedeiro de uma rede que não aquela de onde foi enviada é encaminhada a um retransmissor de correspondência adequado. No presente momento, todas as redes acima são conectadas por retransmissores de correspondência. Assim usuários de qualquer uma delas podem trocar mensagens eletrônicas.

Devido a essa diversidade entre redes e pelo exposto acima, o serviço de *correio eletrônico* em comparação com outros serviços ("login" remoto e transferência de arquivo), é o mais comumente oferecido atualmente pela facilidade de interconexão de redes internacionais. Assim, presentemente, a utilização da facilidade de interconexão de redes internacionais pode ser realmente aproveitada via *correio eletrônico*.

1.3 Objetivo e Apresentação

A implementação do sistema de *correio eletrônico* se depara em perspectiva com o problema da heterogeneidade local: equipamentos de diversos fabricantes, possuindo sistemas operacionais, arquiteturas de rede, protocolos e hardware de comunicação diferentes.

O objetivo deste trabalho é mostrar como se efetuou implementação de correio eletrônico nesse ambiente heterogêneo.

Como para efetivar o alcance do Sistema de Correio Eletrônico é necessário a interação com redes internacionais é analisado no capítulo 2 as características de algumas redes mais proeminentes e num ambiente de interconexão de redes o problema de denominação de recurso do ponto de vista de correio eletrônico.

Para tratar de forma sistemática a implantação do Sistema de Correio Eletrônico são identificadas no capítulo 3 as etapas envolvidas na postagem de uma mensagem, das quais as principais são objeto dos capítulos 4 e 5, tendo em vista a heterogeneidade do ambiente em que foi desenvolvido o sistema descrito no capítulo 6.

A etapa de tradução de denominação e determinação do mecanismo de transporte, identificada no capítulo 3 como das principais, que como o próprio nome diz encaminha a mensagem conforme o seu destino para a rede adequada, é objeto do capítulo 4, em que são descritos os despachadores PMDF e Sendmail.

Como a protocolagem é também uma das principais etapas identificadas no capítulo 3, são analisados no capítulo 5 os conjuntos de protocolos: TCP/IP, DECNET e UUCP.

A descrição do sistema implementado é feita no capítulo 6 através do tratamento dado às etapas identificadas no capítulo 3. Trechos de programas e arquivos de configuração figuram em apêndices.

Capítulo 2

Correio eletrônico e Redes Internacionais

Considerações Iniciais

A facilidade de interconexão de redes permite que usuários tenham acesso além dos recursos do seu sistema de computador e da rede na qual este se conecta, a recursos de redes distintas que atendem às diversificadas necessidades de usuários. Uma facilidade de interconexão implica em serviços básicos oferecidos, denominação de recursos e gerenciamento de rede.

Como o sistema implementado interage com a interconexão de redes internacionais é mostrado na sessão 2.1 as características de algumas redes mais proeminentes do mundo, os principais serviços oferecidos pela facilidade de interconexão dessas redes, os benefícios decorrentes dessa facilidade e exemplos de serviços oferecidos via correio eletrônico. A denominação de recursos num ambiente de interconexão é analisado do ponto de vista de correio eletrônico na sessão 2.2 com exemplos no IFQSC.

2.1 Algumas Redes Proeminentes

Características

Quanto a proposta, administração e financiamento as redes de computadores podem ser classificadas em redes de pesquisa, redes de companhia, redes cooperativas, redes comerciais e *meta-redes*.

Muitas das *principais redes* de computadores foram projetadas e implementadas como *pesquisa* em colocar computadores em rede. Existem ainda um número de redes que ou são elas próprias projetos de pesquisa ou são administradas como apoio a outra pesquisa. Documentação mais detalhada encontra-se em [LJF86], [QH86] e [NS82]. A ARPANET é o exemplo mais conhecido. Tais redes são usualmente administradas por contratadores ou agências de governo, e são mantidas com subsídios do governo. Seus usuários e máquinas hospedeiras não pagam diretamente pelos serviços. Por causa dos seus objetivos e natureza dos seus financiamentos, o acesso a essas redes tende a ser limitado a pesquisadores que participam do trabalho que está sendo financiado.

Existem também um número de *redes militares*, a maioria ou com a mesma tec-

nologia ou estreitamente interoperativas com certas redes de pesquisa. De fato, muitas das *redes de pesquisa* tiveram como um dos seus objetivos o desenvolvimento eventual de *redes militares* correspondentes. O exemplo mais conhecido desse tipo de par é a ARPANET e a MILNET, que estão estreitamente ligadas na **ARPA Internet**. Por causa dessa estreita associação as redes militares são incluídas na categoria de *redes de pesquisa*.

Existem também redes que foram desenvolvidas para prover serviços tipo ARPANET a pessoas e organizações que não poderiam obter acesso a redes de pesquisa e militares financiadas pelo governo, ou que não desejavam estar associadas ao trabalho militar. O exemplo mais conhecido é a CSNET (*Computer Science NETwork*). Usuários dessas redes usualmente pagam uma taxa anual de conexão.

Grandes *companhias* tais como Xerox, DEC, IBM e AT&T têm implementado *redes* internas a fim de dar apoio às suas operações comerciais. Muitas dessas são apenas LAN dentro de prédios particulares, embora algumas sejam de alcance internacional ou mesmo intercontinental. A administração e financiamento de tais redes usualmente provêm de uma única companhia, e seus usuários são na maioria empregados da companhia.

As *redes cooperativas* são redes que cresceram entre comunidades de usuários com interesses similares. Muitas delas, tais como a BITNET (*Because It's Time NETwork*) e suas redes associadas NETNORTH e EARN, se originaram num ambiente acadêmico. Algumas se originaram entre usuários de sistemas de um vendedor particular, exemplo BITNET e IBM, ou de um particular sistema operacional, exemplo UUCP, USENET, EUnet, JUNET, e ACSNET entre usuários UNIX, ou ambos, exemplo FidoNet entre usuários de IBM PC e MS/DOS. Muitos, tais como ACSNET, EUnet, JUNET, UUCP, e USENET possuem usuários com uma mistura de interesses acadêmicos, cooperativos, comerciais e de pesquisa. Frequentemente o vínculo mais forte de uma rede cooperativa particular (pelo menos inicialmente), é o protocolo de rede ou de transporte usado, exemplo RSCS para BITNET, UUCP para a rede de correio UUCP, e FIDO para FidoNet.

A HEPNET (High Energy Physics Network) é uma rede de comunicação de dados que dá suporte à pesquisa em física de alta energia. É basicamente uma rede em forma de estrela de 3 pontas, com o *Lawrence Berkeley Laboratory* e o *Stanford Linear Accelerator* servindo como centro da costa-oeste (Califórnia), o *Fermi National Laboratory* e o *Argonne National Laboratory* servindo como centro do meio-oeste (Illinois), e o *Brookhaven National Laboratory* servindo como centro da costa-leste (N.Y.). Essas instalações representam as instituições de pesquisa nos EUA onde experimentos em alta energia são realmente conduzidos. Juntos formam o “núcleo” de instalação da HEPNET, e as conexões entre eles formam um esqueleto HEPNET de costa a costa. Universidades envolvidas em pesquisa de alta energia dispõem tipicamente de uma linha arrendada para conexão DECNET a uma das instalações do núcleo, a fim de colaborar em experimentos de alta energia.

Também conhecida por HEPNET é a rede européia centrada em torno do CERN (Centre European de Recherche Nucleaire), localizado em Genebra. Como nos EUA, forma-se uma rede de área ampla de instituições e de universidades que estabelecem conexões DECNET com o CERN. Diferentemente dos EUA, entretanto, essas conexões não são de linha arrendada, mas utilizam frequentemente as redes públicas baseadas em X.25 (PSDN).

A SPAN (Space Physics Analysis Network) é uma rede DECNET financiada pela

NASA, que dá suporte à colaboração em pesquisas espaciais e terrestres. Possui quatro roteadores principais, que servem como centros de rede. Estão localizados no *Goddard Space Flight Center*, no *Marshall Space Flight Center* e no *Jet Propulsion Laboratory*. Esses centros roteadores são completamente interconectados. Circuitos saindo desses centros para conexão com universidades e outras instituições envolvidas em pesquisas terrestres e espaciais dão à SPAN uma topologia de multi-estrela.

A administração das redes cooperativas é geralmente distribuída. Algumas, como a BITNET, possuem uma certa quantidade de controle e organização centralizada. Outras, como UUCP (UNIX to UNIX CoPy), funcionam numa quase completa anarquia. As taxas não são geralmente coletadas por uma organização central, mas são pagas por cada nó para conectar-se a outros nós, por exemplo em termos de contas de telefone. Algumas redes cooperativas recebem grandes infusões de dinheiro de companhias específicas. Por exemplo, BITNET tem recebido contribuições massivas da IBM, AT&T gasta mais nas UUCP e USENET do que qualquer outra companhia, e DEC e Philips têm dado grande força para a EUnet.

As *redes comerciais* provêm serviços visando lucro para usuários de fora. Algumas são as bem conhecidas: TYMNET e TELENET (RENPAQ no Brasil). Muitas usam portadoras comuns como o sistema telefônico. A administração é sempre centralizada, embora a execução possa ser delegada. Os fundos usualmente são derivados das taxas cobradas a pessoas individuais ou organizações por tempo de conexão de CPU. A maioria das redes comerciais não permite livre troca de correspondência com outras redes.

Existem vários projetos sendo dirigidos no sentido de estender a comunidade de usuários de rede conectando as redes existentes e as ainda não construídas em *Meta-redes*. Tais redes diferem das inter-redes nos seus objetivos e nas suas partes constituintes que freqüentemente possuem protocolos dissimilares a nível tão alto quanto a camada de transporte. A maioria das *meta-redes* ainda não existem, a CSNET é uma exceção. A NSFnet é uma iniciativa recente (1987) da National Science Foundation no sentido de provêr uma estrutura para uma Rede de Pesquisa Acadêmica Nacional (National Academic Research Net). A principal motivação para a atividade da NSFnet (experiências técnicas com as redes já existentes, busca de soluções para meio físico cada vez mais rápido) é prover acesso de alta qualidade a recursos de supercomputadores à comunidade de engenheiros e cientistas o mais cedo possível.

Várias redes não se encaixam inteiramente em nenhuma dessas categorias. A COSAC, JUNET e SDN poderiam ser classificadas como de *pesquisa* ou *cooperativas*. COSAC é enquadrada como de pesquisa por causa da sua relação com as redes EAN e DFN. SDN e JUNET são agrupadas com redes cooperativas por causa da sua relação com UUCP, USENET, e EUnet.

A tabela 2.1, retirada de [QH86], mostra as características de algumas redes proeminentes com relação ao país centro, à extensão que é indicado por um par de números dos continentes e nações que alcança, o número de hospedeiros, usuários, camadas de rede e transporte, serviços oferecidos e qualidade.

Os protocolos das camadas de rede indicados nessa tabela podem ser:

- a:** subrede de comunicação estilo ARPANET PSN (Packet Switch Node - BBN 1822)
- d:** linha telefônica discada
- e:** Ethernet

i: uma inter-rede sobre várias camadas de rede

p: linha telefônica arrendada

x: X.25 (usualmente sobre linha telefônica arrendada)

Os protocolos das camadas de inter-rede e transporte indicados na tabela podem ser:

A: SUN III da ACSNET

B: "Coloured Book" da JANET

C: CSNET: TCP/IP (ARPANET); TCP/IP EM X.25 (X25NET); MMDF2 (Phonenet)

D: DECNET da DEC

F: FidoNet

N: Conjunto de protocolos para serviços de rede da Xerox

P: Pacote Universal de Protocolo (PUP) da PARC Xerox

T: o conjunto de protocolos da ARPA TCP/IP

U: UUCP da AT&T

R: RSCS da IBM

X: CCITT/ISO X.400 e protocolos relacionados

Os serviços podem ser:

l: "login" remoto

f: transferência de arquivo

m: correio eletrônico

n: notícias

o: outros

A qualidade é uma tripla que expressa velocidade em bits por segundo, tempo médio de entrega das mensagens no serviço de correio eletrônico que pode ser *m* para minutos *h* para uma hora ou mais e *d* para um dia ou mais, e confiabilidade que na escala subjetiva de Quarteman e Hoskins vai de 1 a 10.

A fig.2.1 ilustra a extensão e heterogeneidade da Internet. O sistema é constituído por redes terrestres tais como Arpanet e redes públicas X.25, várias fases de redes de pacotes de rádio, um número de redes locais, e duas redes distintas de comutação de pacotes via satélite. Grosseiramente 100 redes e 60 entroncamentos (1985), todos interconectados num sistema unificado, provêm serviço de comunicação de computadores de forma robusta e confiável requerida por usuários militares e comerciais. A coordenação da Internet é provida pelo NIC (Network Information Center) no BBN (Bolt, Beranek and Neuman).

A fig.2.2 ilustra o tráfego de pacotes da NSFnet de 1987 a 1990. Pode-se notar a tendência de um crescimento exponencial. Esse fato não deve ser muito diferente para outras redes.

Tabela 2.1: Características de algumas Redes Proeminentes

Nome	Centro	Extensão	Hosp.	Usuários	Camadas Rede	Transp.	Serviços	Qualidade
Redes de Pesquisa								
ARPA Internet	EUA	3,8	2.050	?	i	T	lfmno	56.000,m,9
ARPANET	EUA	1,1	150	?	a	T	lfmo	56.000,m,9
MILNET	EUA	2,3	400	?	a	T	lfmo	56.000,m,9
MINET	Europa	1,4	?	?	x	T	lfmo	9.600,m,8
CSNET	EUA	4,10	170	?	i	C	m	1.200,h,8
Phonet	EUA	4	128	?	d	M	m	1.200,h,7
X25NET	EUA	1	18	?	x	T	lfu	9.600,m,8
ARPANET (CSNET hosts)	EUA	1,1	25	?	i	T	lfmo	56.000,m,9
Cypress	EUA	1	6	?	p	T	lfmo	9.600,m,9
NSFNET	EUA	1,1	65	?	i	T	lfmo	T1,m,9
MFENET	EUA	1,1	120	?	i	D	fu	56.000,?,?
SPAN	EUA	1,1	100+	?	px	D	fu	56.000,m,8
MAILNET	EUA	2,3?	28	1.800	dx	M	m	1.200,h,9
JANET	Reino Unido	1,1	915	?	x	B	lfmo	4.800,?,?
EEAN networks	Europa	3,12	33	?	x	X	m	2.400,m,9
CDNnet	Canadá	1,1	32	?	x	X	lfmo	2.400,m,9
COSAC	França	1,2	27	?	x	X	fm	1.200,d,7
DFN	Alemanha	1,1	6	?	x	X	fm	9.600,?,?
Redes de Companhia								
Xerox Internet	EUA	3,4	?	12.000	i	N	lfmo	56.000,h,9
Xerox RIN	EUA	1	?	4.000	i	NPT	lfmo	56.000,h,9
Xerox CIN	EUA	3	?	8.000	i	N	lfmo	56.000,h,9
DEC's Easynet	EUA	4,20?	10.000+	60.000?	d	D	lfmo	56.000,m,9
IBM's VNET	EUA	4,?	2.200	?	p	R	fmo	9.600,h,9
Redes Cooperativas								
BITNET	EUA	3,21	1.306	?	p	R	fmo	9.600,h,8
BITNET	EUA	1,2	845	?	p	R	fmo	9.600,h,8
NETNORTH	Canadá	1,1	91	?	p	R	fmo	9.600,h,8
EARN	Europa	1,17	363	?	px	R	fmo	9.600,h,8
Asianet	Japão	1,1	7	?	p	R	fmo	9.600,h,8
Fidonet	EUA	2,4?	500	?	d	F	m	1.200,d,4
ACSNET	Austrália	1,1	300	?	dep	A	fmno	1.200,h,9
UUCP mail	América do Norte	4,5	7.000+	200.000?	dx	U	m	1.200,d,5
USENET news	América do Norte	4,4	2.500+	50.000?	dx	UATR	n	1.200,-,7
EUnet	Europa	1,13	896	?	dpx	UAT	lfmo	1.200,h,8
SDN	Coreia	1,1	100	?	epx	TU	lfmo	2.400,d,8
JUNET	Japão	1,1	160	?	x	U	mn	2.400,d,8

Figura 2.1: Sistema Heterogêneo que constitui a DARPA Internet

Packet Traffic on NSFNET

Figura 2.2: Tráfego de Pacotes na NSFnet

Serviços Básicos Oferecidos

Os serviços básicos mais comuns oferecidos hoje por redes de comunicação de computadores são correio eletrônico, transferência de arquivo, acesso remoto a sistema, e acesso a base de dados.

Correio eletrônico é o serviço de rede mais comumente disponível. No caso mais simples, um usuário compõe uma mensagem num computador local, fornece o endereço da caixa de correio do receptor, que pode estar em um computador remoto; e instrui o sistema de correio local para enviar a mensagem ao receptor indicado. O sistema de correio local então assume a responsabilidade da mensagem. A mensagem é enviada via rede ao hospedeiro destino e eventualmente aparece na caixa de correio do receptor. Se o receptor é uma lista de receptores ao invés de um individual, o sistema de correio destino encaminhará cópias da mensagem ao membros da lista. A maioria dos sistemas de correio modernos ou assegurarão a entrega das mensagens ou eventualmente proverão uma indicação de erro ao que enviou, por exemplo, a caixa de correio não existe, o hospedeiro destino caiu. A natureza assíncrona da comunicação de correio eletrônico permite aos usuários ler e responder as mensagens conforme as suas conveniências. Junto com a rápida entrega das mensagens, esta forma assíncrona conduz a uma comunicação muito mais eficiente do que é possível via correio postal ou sistema telefônico.

O serviço de *transferência de arquivo* é usado para transferir arquivos entre dois hospedeiros de rede. O usuário que inicia a transferência deve possuir os direitos de acesso aos arquivos envolvidos. A transferência de arquivo trabalha melhor quando os arquivos fonte e destino são organizados de forma similar e usam as mesmas convenções de codificação de dados. Simples arquivos seqüenciais podem geralmente ser transferidos com pouca dificuldade. Arquivos estruturados, por exemplo indexados, indexados de forma seqüencial, de acesso aleatório, cujas organizações dependem dos detalhes de um sistema de arquivo particular podem geralmente ser transferidos somente entre hospedeiros que suportam sistemas de arquivos similares.

O serviço de *acesso remoto a sistema* ou "*login*" *remoto* permite que um usuário trabalhando num computador hospedeiro conecte-se e faça "*login*" em outro computador hospedeiro da rede sem desconectar do primeiro hospedeiro. O usuário pode iniciar o trabalho no computador remoto e mais tarde retornar e continuar a sessão no hospedeiro original. Essa facilidade é conveniente se alguém rotineiramente acessa mais do que um computador na rede. Por exemplo, um usuário de um minicomputador local pode desejar periodicamente fazer "*login*" num supercomputador CRAY remoto. Usuários devem possuir privilégios de "*login*" para todos os hospedeiros envolvidos. Os protocolos de "*login*" remoto trabalham melhor quando os dois sistemas são similares com respeito a interface de sessão de terminal apresentada aos usuários. Por exemplo, especial cuidado deve ser tomado a fim de capacitar um usuário de um sistema operacional orientado a linha tal como o VAX/VMS obter serviço orientado a tela quando acessa um sistema IBM via um protocolo de "*login*" remoto e vice-versa.

As redes provêm, às vezes, *bases de dados* centrais que se pode ter acesso em hospedeiros da rede. Uma tal base de dados pode ser um guia, que contém informações de usuários da rede, tais como endereço de correio eletrônico e número de telefone. Por exemplo, a fim de dar apoio aos usuários na busca de endereços de correio corretos a CSNET provê um sistema base de dados, servidor de nomes. Dadas palavras chaves descrevendo um receptor de correio, o servidor de nome retorna o endereço de correio

do(s) indivíduo(s) cuja descrição corresponde na base de dados central com as palavras chaves. A ARPANET e a BITNET oferecem um serviço similar. Serviços desse tipo se tornarão mais importantes a medida que redes adicionais são interconectadas a fim de formar inter-redes complexas.

Outro exemplo de uma base de dados baseada em rede é a coleção de relatórios técnicos (conhecidos como *RFCs*) mantidos na ARPANET. Usuários de hospedeiros conectados a qualquer rede que é parte da *Department of Defense (DoD) Internet* pode obter arquivos dessa base de dados. Bases de dados similares são mantidas pela BITNET e CSNET.

No futuro um número de serviços adicionais se integrarão. Um exemplo é a transferência de dados multi-meio, onde voz, texto e gráfico possam ser misturados numa mensagem única. Também uma rede de grande largura de banda será capaz de prover transferência de dados de alta velocidade indo e vindo de supercomputadores ou capacitando transmissão de imagens gráficas em tempo real (ou via satélite ou linhas de alta velocidade por terra). Serviços de conferência baseados em computador poderão se tornar mais largamente disponíveis.

Benefícios de Redes de Computador

Uma rede de computador que provê os serviços básicos acima citados, oferece numerosos benefícios a engenheiros e cientistas, tais como aumento de oportunidade de colaboração com colegas, compartilhamento de recursos de computador, rápida disseminação de relatórios e informação, desenvolvimento de grupos de interesse especial, e desenvolvimento de serviços especializados. Deve-se notar que esses benefícios não são hipotéticos mas já estão disponíveis a um grande número de indivíduos atualmente usando redes de computador. A seguir serão apresentados os benefícios com relação a cada serviço oferecido por redes.

Correio eletrônico, transferência de arquivo, e acesso a sistema remoto todos facilitam a colaboração entre indivíduos geograficamente dispersos tendo acesso a um hospedeiro de rede. Colaboração desse tipo já é comum em locais onde o computador já é utilizado como ferramenta de trabalho, como por exemplo entre pesquisadores de Física e Ciências de Computação em universidades dos Estados Unidos e entre funcionários de grandes companhias tais como DEC e IBM.

Com algumas redes, por exemplo ARPANET, o componente X.25 da CSNET (X25.net) ou BITNET, a transferência de mensagens é feita normalmente após a mensagem ser enviada pelo usuário, e é possível conduzir conversações por correio eletrônico com atrasos de poucos segundos ou poucos minutos na entrega de mensagens aos receptores. Em outras redes tais como CSNET e Phonenet o atraso das transmissões de mensagem pode alcançar horas ou dias. Alguém pode enviar comentários (documentos, programas, idéias) quando estão prontos com a certeza que eles logo serão vistos pelo receptor. A lerdeza ou inconfiabilidade do serviço postal e as dificuldades de alcançar indivíduos ocupados por telefone são todos evitados.

Transferência de arquivo pode ser usada para enviar e recuperar manuscritos, códigos, ou outro material capaz de ser lido por máquinas preparado por colaboradores e a eles destinados. Isso torna fácil que colaboradores em diferentes lugares reünam trabalho na produção de manuscritos ou sistemas de programas, por exemplo o trabalho completado a cada dia pode ser transferido a um repositório central e então distribuído a todos os

participantes do projeto.

Acesso a sistema remoto capacita colaboradores de diferentes lugares usarem o mesmo sistema para os seus serviços. Em ciências de computação isso é particularmente importante quando um programa (*software*) está sendo desenvolvido por um grupo (possivelmente disperso). Existem analogias em qualquer campo onde projetos de multi-investigação e de múltiplos lugares são usuais. O melhor exemplo é provavelmente o acesso a supercomputadores, que são máquinas cujo custo e potência computacional excedem aquelas que podem ser justificadas pela maioria das organizações. A *National Science Foundation* está atualmente construindo a NSFnet a fim de prover tal acesso a pesquisadores. Outro exemplo é a proposta de compartilhar o telescópio espacial da NASA via uma rede.

Redes, com o uso de correio eletrônico, podem suportar a formação de grupos de interesse especial cujos membros se comunicam sobre um tópico particular. Listas de postagem para tais grupos de interesse especial são extremamente comuns nas redes de computadores existentes. Os tópicos de interesse de tais grupos podem variar de um particular microcomputador às implicações filosóficas da pesquisa em inteligência artificial. Existem centenas, senão milhares, de tais listas de postagem de grupos de interesse disponíveis nas redes ARPANET, BITNET, CSNET, E UUCP. Essas listas de postagem de grupos de interesse geralmente utilizam o sistema de correio de um único hospedeiro para remeter mensagens a todos endereçados na lista. A fim de reduzir o volume de tal tráfego, as mensagens são freqüentemente moderadas, ou editadas, e então coletadas em *compêndios* antes de serem remetidas. Além disso, hospedeiros locais freqüentemente mantêm uma lista de postagem local dos mais populares grupos de interesse para redistribuição, de maneira que apenas uma cópia de cada mensagem necessita ser enviada ao hospedeiro.

Correio Eletrônico e/ou Transferência de arquivo podem ser empregados a fim de prover serviços especiais aos usuários de rede. Um exemplo é a facilidade de fabricação de VLSI chamado MOSIS. Os pesquisadores com acesso à ARPANET ou CSNET podem enviar projetos de VLSI via rede para o *University of Southern California Information Sciences Institute* onde eles são processados e enviados para fabricação. Semanas mais tarde *chips* completos são enviados para o pesquisador para teste e experiência. O sistema é altamente eficiente e acelera imensamente o tempo de ida e volta para recepção dos *chips* finais. Outro exemplo é a extração de software de domínio público. A divulgação e acesso através desse serviço ajudou a aceitação de diversos padrões, como por exemplo: Kermit, Xwindows e TCP/IP.

Facilidades via Correio Eletrônico

Como o serviço de *correio eletrônico* é o mais comum, já existem servidores para atender a transferência de arquivo via *correio eletrônico* e execução em "batch" de um processo ativado por mensagens. Programas executáveis são transformados de binário para ascii e já existem *softwares* nos servidores que dividem arquivos grandes em vários arquivos para não exceder o tamanho das mensagens. O próprio usuário pode se utilizar desse recurso usando os softwares públicos encontrados nas próprias bases, tais como: BtoA e UUCODE, (codificações Binário $\leftrightarrow$ ASCII), e MailSplit, para dividir (e depois reunir) arquivos grandes a serem enviados como mensagens por *correio eletrônico*.

Exemplos desses servidores seriam:

Listserv@RPI ECS (Rensselaer Polytechnic Institute)

Listserv@NDSUVMI (North Dakota State University)

Servidores da BITNET que enviam por correio a solicitação, também por correio a eles destinada, de porções selecionadas da base SIMTEL20 de software de domínio público.

Entre esses, os arquivos dos diretórios:

PD:<CPM.*>, PD:<SIGM.*>, PD:<PCBLUE.*>, PD:<MSDOS.*>, PD:<MISC.*>

O arquivo 000-MASTER-INDEX.TXT no diretório PD:<CPM> possui o índice dos arquivos no diretório e uma pequena descrição de cada um. Esse arquivo existe nos vários diretórios e é bastante útil.

Os comandos disponíveis são /PDDIR para listar arquivos e /PDGET para solicitar um arquivo específico.

Exemplo de uma solicitação:

MAIL

TO: in%"LISTSERV@RPI ECS"

SUBJECT:

/PDGET HELP

/PDGET PD:<diretório.subdiretório>arquivo

----- fim da mensagem -----

Esse arquivo viria codificado com uuencode (binário->ascii) e o usuário teria que decodificar com uudecode (ascii->binário).

Um exemplo de servidor que recebe solicitação pelo campo assunto é o SERVICE@NIC.DDN.MIL da Internet.

Exemplo de solicitação:

MAIL

TO: in%"SERVICE.NIC.DDN.MIL"

SUBJECT:HELP (* para dar informação sobre o servidor *)

ou

SUBJECT: RFC### (* para solicitar o envio de um documento

RFC, exemplo: RFC987 *)

Outro exemplo seria o NEWS-ARCHIVE@SUN.COM da BITNET, que dá as novidades de equipamentos SUN.

Outro exemplo de servidores que recebem solicitações por campos determinados dentro da mensagem é o INFO-SERVER@SH.CS.NET servidor da CSNET.

Exemplo de solicitação:

MAIL

TO: in%"INFO-SERVER@SH.CS.NET"

SUBJECT:


```
REQUEST: RFC
TOPIC: RFC####
REQUEST: END
```

onde os REQUESTs poderiam ser

```
CALENDAR
FTP
INFO
CREN
NSFNET
RFC
MOD-SOURCES
```

Em particular, para solicitar um fonte de software público, a mensagem seria:

```
REQUEST: MOD-SOURCES
TOPIC: fonte-solicitada
REQUEST: END
```

2.2 Denominação de Recurso e Exemplos no IFQSC

Nome, Endereço e Rota

Do ponto de vista do usuário, o que é necessário para enviar correio eletrônico? Uma ferramenta que ajude a compor a mensagem e o conhecimento pelo usuário do “endereço” do destinatário.

Na idéia de especificação de destinatário surgem três termos relacionados - nome, endereço e rota - [Sho78], que são frequentemente confundidos, não só em termos de correio eletrônico, mas de especificação de recurso em rede. O nome de um hospedeiro, caixa postal ou outro recurso, é o que um usuário usa para indicar o recurso desejado. Seu endereço especifica a localização do recurso para o software de rede. Uma rota é usada por um software de rede afim de determinar como conseguir o recurso.

Numa analogia com a rede telefônica, um nome é um nome pessoal, tal como João Silva, um endereço é um número de telefone, e uma rota é uma sequência de linhas telefônicas e mudanças que são feitas para atingir o número do João pelo telefone que chamou.

No ambiente de interconexão de redes a coisa não é tão simples quanto no sistema telefônico. No sistema telefônico existe uma lista de assinantes que mapeia o nome no endereço. Essa lista é organizada por regiões geográficas e existe um padrão aceito internacionalmente em que o número de telefone conseguido é acessado por qualquer lugar do mundo, colocando os códigos de área correspondentes como prefixos. Considerando em alto nível a colocação dos códigos de área seria a interferência do usuário na rota.

No ambiente de interconexão de rede existem várias redes que foram constituídas independentemente. Cada uma com suas próprias motivações; organizações políticas e

geográficas; tendência a certos fabricantes e sistemas operacionais; arquiteturas, protocolos e hardware de rede. Essa diversidade faz com que não só o mapeamento nome-endereço, mas também a denominação do recurso não seja imediata. Um endereço válido para uma rede não significa nada para outra. Além disso, como denominar um recurso? Pelo nome da rede, pela região, pela motivação da rede, pelas subdivisões políticas? Isso levou a idéia de que para denominação teria-se que identificar conjuntos e subconjuntos de maneira recursiva, utilizando algum critério comum, de forma a que pelo menos a denominação do recurso fosse algo padrão. E o mapeamento nome-endereço? Deixa que o computador faça a pesquisa numa tabela. Mas manter a tabela numa única máquina é inviável, pelo tamanho, manutenção e concorrência na pesquisa. Isso leva a pensar na distribuição das tabelas de mapeamento nome-endereço por várias máquinas. Mas como fazer essa distribuição, já que isso influencia as rotas, disponibilidade de recursos e certa política.

Para solucionar isso a ARPA Internet adotou o sistema de domínios de nomes [Moc87a] [Par86] [Sta87] [Lot87] [Moc87b]. Esse sistema é uma tentativa de descentralizar a administração do mapeamento de nomes em endereços de hospedeiros pelo uso de servidor de nomes. Cada servidor controla parte do espaço de nomes. Na Internet isso tornou-se necessário, parte porque a tabela estática de hospedeiros usada primeiramente para esse propósito tem se tornado inviável com o crescimento da Internet. E parte porque a maioria dos hospedeiros da Internet estão em redes locais de organizações particulares sendo desejável, assim, permitir que a organização local controle o mapeamento. O sistema de domínio de nomes também implementa um sistema hierárquico de denominação e provê protocolos para comunicação com o servidor de nomes.

Um domínio, portanto, é um agrupamento administrativo. Cada domínio é responsável pela manutenção da informação nos subdomínios. Cada subdomínio, por sua vez, sendo um agrupamento administrativo com direitos próprios.

Os nomes com estilo domínio são da forma

`hospedeiro.domínio1...domínioN`

Quatro são os domínios de mais alto nível (domínio N) que têm sido definidos: “edu”, “com”, “mil”, e “gov” que representam respectivamente instituições educacionais, empresas comerciais, organizações militares e agências de governo. Fugindo da proposta inicial, mas para conseguir denominar vários hospedeiros, surgiram domínios para países como UK, AU, BR e o domínio NET para designar as organizações administrativas de redes tais como CSNET, UUCP e BITNET.

Exemplos de subdomínios com domínios *edu* são “wisc”, “stanford”, e “cornell”. A universidade de Wisconsin é responsável pelos hospedeiros no subdomínio *wisc*. Assim a máquina *rsch* em um nome estilo domínio é da forma “rsch.wisc.edu”.

Para cada domínio, existe uma base de dados que inclui o endereço dos servidores de nomes para os subdomínios e/ou os endereços dos hospedeiros dentro do domínio. Cada domínio tem pelo menos dois servidores de nomes em diferentes localizações na interrede. Segundo o apresentado em [Par86], quando um despachador de correspondência recebe uma solicitação para enviar uma mensagem com um nome estilo domínio, ele interroga um *servidor de nomes com domínio* para aquele domínio. Enquanto o protocolo real permite uma variedade de respostas, no caso mais simples, o *servidor de nomes com*

domínio interrogado, retorna ou um endereço do *servidor de nomes* para o próximo nível de domínio (mais baixo no nome estilo domínio) ou o endereço do hospedeiro destino. No primeiro caso, o despachador de correspondência então, interroga o *servidor de nomes com domínio* cujo endereço acabou de receber. O processo termina quando o despachador de correspondência obtém o endereço "Internet" do hospedeiro que se deseja enviar a correspondência. Mapeamentos adicionais desse endereço "Internet" a endereços de entroncamentos é manipulado a um nível mais baixo.

A rede britânica JANET tem um sistema de domínio similar ao da ARPA Internet, porém com os domínios em ordem oposta na denominação hierárquica. A raiz (domínio mais externo, ou de nível mais alto) fica mais à esquerda. A rede australiana ACSNET também possui um sistema de nomes com domínio tipo o da Internet.

A rede UUCP usa para denominação do recurso, a rota para atingi-lo. Exemplo: hospedeiro1!hosp2!...hospN!usuário. Esse tipo de denominação é chamado encaminhamento pela fonte. Quando o software de rede determina a rota, o encaminhamento é chamado pelo sistema. As meta-redes de diferentes redes e inter-redes, frequentemente, requerem encaminhamento pela fonte afim de alcançar a rede apropriada, porque não existe ainda uma convenção de denominação universalmente aceita. Encaminhamentos fontes como alpha!beta%gama@delta são infelizmente ainda comuns.

O padrão de correspondência ISO X.400 possui uma denominação hierárquica que usa atributos [Ott87] [R+86] [Cun83] [Cun84]. Os conflitos de nomes podem ser resolvidos pela suficiente especificação de atributos. A especificação contida em [Kil86] foi recentemente formulada para uma tradução entre a denominação hierárquica com domínios da ARPA Internet e a denominação hierárquica de atributos do X.400. As redes EAN baseadas no X.400 usam uma versão simplificada do sistema X.400, pois não existe ainda nenhum registro para domínios do X.400.

Em um encontro recente (1986), os representantes norte-americanos da ARPA Internet, BITNET, CSNET e UUCP decidiram adotar a sintaxe de denominação hierárquica com domínios da ARPA Internet e os próprios domínios como uma sintaxe de denominação comum. A adoção por cada hospedeiro é voluntária na UUCP e BITNET. A EUnet na Europa está indo na mesma direção, e já tem registrado vários domínios nacionais de nível mais alto. A JUNET no Japão tem um sistema de domínios similar. A meta-rede NSFnet tem também se padronizado nos protocolos TCP/IP e relacionados [L+85] [Pad83]. Assim os domínios Internet podem tornar-se padrão de fato, pelo menos nos Estados Unidos.

Porém, enquanto isso não acontece, uma denominação que possa ser utilizada nos vários sistemas de denominação hierárquica atualmente em uso por essas redes, ainda é um problema. Alguns sistemas têm limite para o número de caracteres do nome da máquina ou usuário. Exemplo: para a BITNET o nome da máquina e usuário é até oito caracteres, para o UUCP o nome da máquina é até sete caracteres, para DECNET a máquina é até seis caracteres. Além disso não há limitação teórica para o número de domínios tanto Internet quanto X.400, mas existe limitação prática de alguns softwares. Exemplo disso é a versão experimental do EAN (X.400) que temos no VAX do IFQSC que limita a seis o número de domínios.

Só para ilustrar, vejamos uma denominação escrita para várias redes.

steve@cs.ucl.ac.uk
...!ucl-cs!steve

via ARPA Internet
via UUCP

<code>steve@uk.ac.ucl.cs</code>	via JANET
<code>gb/bt/des/steve(ucl/cs)</code>	X.400, GIPSI (do INRIA) UIP
<code>/C=GB/ADMD=BT/PRDM=DES/O=UCL/</code>	
<code>OU=CS/S=Kille/</code>	X.400, RFC987 UIP
<code><C=gb;A=bt;P=des;O=ucl;S=steve;OU=cs></code>	X.400, outro UIP
<code>steve!ucl!cs&des%bt&gb</code>	X.400, DFN UIP
<code>steve!ucl!cs#des&bt.gb</code>	X.400, EAR/X.400 gateway UIP
 <code>steve@cs.ucl.bt.gb</code>	 EAN, RFC822 UIP e outro dominio
<code>/C=/ADMD=/PRDM=UK/DD.=cs.ucl.ac/</code>	EAN, codificacao X.400, RFC987 UIP
<code>DD.=steve</code>	

Uma abordagem para interconexão de redes completamente distintas é o uso de entroncamentos [H⁺83] [BE83] [Sch83] [RW83] [IW87]. Esses entroncamentos são hospedeiros ou processadores especiais que são conectados a duas ou mais redes e que entendem os protocolos de cada uma delas. Muitos entroncamentos ou retransmissores operam em níveis baixos de protocolos e são transparentes aos usuários. No entanto, entroncamentos entre redes com diferenças a nível da camada inter-rede, operam de forma menos transparente ao usuário. Além disso o serviço de *correio eletrônico*, frequentemente, é o único oferecido. Infelizmente os usuários necessitam para a denominação do destinatário saber a topologia da inter-rede ou as convenções particulares da rede destino. Geralmente a questão é resolvida com denominações da forma “usuário%hospedeiro@retransmissor-de-correspondência”, onde o “usuário%hospedeiro” especifica corretamente o endereço do receptor na rede destino e “retransmissor-de-correspondência” a denominação da máquina retransmissora na rede fonte.

Exemplos no IFQSC

Um exemplo disso é a denominação que usávamos antes da FAPESP ser retransmissor de correspondência. A FAPESP era um elo transparente a nível de usuário que nos conectava com o exterior.

MAIL

TO: LBL::"nome-destino"

ou

TO: CSA::"nome-destino"

Onde LBL e CSA na sintaxe de nossa máquina fonte eram as máquinas retransmissoras de correio para onde a mensagem com o “nome-destino” deveria ser passada para ser por eles encaminhada.

Em alguns casos os retransmissores podem não ser conhecidos, em outros não é possível revelá-los por considerações políticas ou econômicas. A tabela 2.2 retirada de [QH86] é uma compilação das sintaxes de encaminhamento de correspondência entre algumas redes. Contudo esses entroncamentos estão sujeitos a mudança e a natureza da informação torna impossível compilar uma tabela que seja válida por muito tempo.

Tabela 2.2: Sintaxes de encaminhamento de correspondências entre redes

Domínios:		I: com,edu,etc.		
De	para:	ARPA Internet	CSNET Phonenet	MAILNET
ARPA Internet		<i>v@d.I</i>	<i>v%h.csnet@relay.cs.net</i>	<i>u%h.mailnet@mit-multics.arpa</i>
CSNET Phonenet		<i>u@d.I</i>	<i>v@d.I</i>	<i>u%h.mailnet@mit-multics.arpa</i>
MAILNET		<i>u% d.I@mit-multics</i>	<i>u% d.I%relay.cs.net@mit-multics</i>	?
JANET		<i>v% d.I@uk.ac.ucl.cs</i>	?	?
EAN		<i>u@d.I</i>	<i>u@d.I</i>	<i>u@h.mailnet</i>
COSAC		<i>adi/u% d.I@relay.cs.net</i>	<i>adi/u% d.I@relay.cs.net</i>	?
BITNET		<i>v@d.I</i>	<i>u@h.csnet</i>	<i>u@h.mailnet</i>
ACSNET		<i>u% d.I@munnari.oz</i>	<i>u% d.I@munnari.oz</i>	<i>u%h.mailnet@munnari.oz</i>
UUCP		<i>e!d!u</i>	<i>e!d!u</i>	<i>e!h.mailnet!u</i>
JUNET		<i>u@d.I.arpa</i>	<i>u@d.I.arpa</i>	<i>u%h.mailnet@mit-multics.arpa</i>
Domínios:		U: uk	E: cdn, dfn, etc.	
De	para:	JANET	EAN	COSAC
ARPA Internet		<i>v% d.U@cs.ucl.ac.uk</i>	<i>u% d.E@ubc.csnet@relay.cs.net</i>	<i>h/u%france.csnet@relay.cs.net</i>
CSNET Phonenet		<i>u% d.U@cs.ucl.ac.uk</i>	<i>u% d.E@ubc.csnet</i>	<i>h/u@france.csnet</i>
MAILNET		?	<i>u% d.E@ubc.mailnet</i>	?
JANET		<i>v@U.d</i>	<i>v@d.U</i>	?
EAN		<i>u@d.U</i>	<i>u@d.E</i>	<i>h/u@france.csnet</i>
COSAC		?	?	<i>h/u</i>
BITNET		<i>u% d.U@ac.uk</i>	<i>u@d.E</i>	<i>h/u@france.csnet</i>
ACSNET		<i>u% d.U@munnari.oz</i>	<i>u% d.E@munnari.oz</i>	<i>h/u%france.csnet@munnari.oz</i>
UUCP		<i>e!cs.ucl.ac.uk!d.U!u</i>	<i>e!d.E!u</i>	?
JUNET		<i>v@d.U.janet</i>	<i>u% d.E@ubc.csnet</i>	<i>h/u@france.csnet</i>
Domínios:		R: A registry		
De	para:	Xerox Internet	DEC's Easynet	IBM's VNET
ARPA Internet		<i>u.R@xerox.com</i>	<i>u%h.dec@decwrl.dec.com</i>	<i>u%h@ibm.com</i>
CSNET Phonenet		<i>u.R@xerox.com</i>	<i>u%h.dec@decwrl.dec.com</i>	<i>u%h@ibm.com</i>
MAILNET		?	?	?
JANET		?	?	?
EAN		<i>u.R@xerox.com</i>	<i>u%h.dec@decwrl.dec.com</i>	<i>u%h@ibm.com</i>
COSAC		?	?	?
BITNET		<i>u.R@xerox.com</i>	<i>u%h.dec.com@decwrl.dec.com</i>	<i>u@vnet</i>
ACSNET		<i>u.R% xerox.com@munnari.oz</i>	<i>u%h.dec.com@munnari.oz</i>	<i>u%h@ibm.com@munnari.oz</i>
UUCP		<i>parcvax!u.R</i>	<i>decwrl!h.dec.com!u</i>	<i>e!ibm.com!u%h</i>
JUNET		<i>u.R@xerox.com.arpa</i>	<i>u%h.dec@decwrl.dec.com.arpa</i>	<i>u%h@ibm.com.arpa</i>
Domínios:		A: oz.au	J: junet	
De	para:	BITNET	UUCP	JUNET
ARPA Internet		<i>u%h.bitnet@wiscvm.wisc.edu</i>	<i>u%h.uucp@c</i>	<i>u% d.J%tokyo-rela</i>
CSNET Phonenet		<i>u%h.bitnet@relay.cs.net</i>	<i>u%h.uucp@c</i>	<i>u% d.J%tokyo-rela</i>
MAILNET		?	<i>u% d.A% c@mit-multics</i>	<i>u% d.J%csnet-relay</i>
JANET		?	<i>u% d.oz@uk.ac.ukc</i>	?
EAN		<i>u@h.bitnet</i>	<i>u@d.A</i>	<i>u% d.J@uk.ac.ukc</i>
COSAC		<i>adi/u%h.bitnet@relay.cs.net</i>	?	<i>u% d.J@relay.cs.net</i>
BITNET		<i>u@h</i>	<i>h1!h2!h!u@psuvax1</i>	<i>u% d.J@csnet-relay</i>
ACSNET		<i>u%h.bitnet@munnari.oz</i>	<i>u@d.A</i>	<i>u% d.J@munnari.oz</i>
UUCP		<i>psuvax1!h.bitnet!u</i>	<i>seismo@munnari!d.A!u</i>	<i>e!d.J!u</i>
JUNET		<i>u@h.bitnet</i>	<i>u@d.A</i>	<i>u@d.J</i>

Notas: *ubc-can* é um entroncamento de UUCP para CDNnet; de EUnet para as redes EAN europeias existe um entroncamento por país; existe mais de um entroncamento entre BITNET e UUCP; UUCP, EUnet, e SDN são endereçadas similarmente, assim apenas uma delas está listada aqui.

Abreviações: *u*: usuário, *h*: hospedeiro, *c*: entroncamento (não nomeado aqui); *d*: domínio.

Omissões: de redes de companhia, com redes comerciais, com a ARPA Internet

Note-se que esse encaminhamento pela fonte depende das sintaxes fonte, retransmissor e destino e o encadeamento de vários ou mistura com a sintaxe do UUCP podem resultar em problemas de análise ("parsing") nos retransmissores de correspondência.

Em geral, os hospedeiros retransmissores possuem softwares despachadores de correspondência que se incumbem de compor o nome-destino com o nome-do-retransmissor adequado e submetem a correspondência para ser transmitida à rede adequada do retransmissor. No nosso exemplo o LBL e CSA já possuíam despachadores de correspondência.

Quando a FAPESP adquiriu o software despachador PMDF (Pascal Memo Distribution File), passamos a usá-la ao invés das máquinas LBL e CSA. Exemplo:

MAIL

TO: FPSP::IN%"nome-destino"

onde IN% indica que se deve usar o despachador PMDF

Quando nós próprios adquirimos o despachador, não era mais necessário o usuário se preocupar em repassar para FAPESP para lá ser despachada a mensagem. Assim, ficou:

MAIL

TO: IN%"nome-destino"

A dificuldade para denominação no Brasil é ainda mais acentuada. As ligações em rede estão sendo feitas por iniciativas próprias à FAPESP e segundo a disponibilidade de acesso, o que torna a configuração político-administrativa ainda mais caótica. Isso fez com que a FAPESP tivesse problemas na designação dos nomes a essas máquinas, tanto do ponto de vista político quanto econômico. Se uma máquina usa um meio de transmissão mais caro que outra e essa está na rota de outras, quem é que paga? E são coisas que não são e não devem ser resolvidas burocraticamente e sim automaticamente pela escolha adequada dos domínios para a denominação hierárquica. Exemplo, a denominação do VAX do IFQSC é "ifqsc.usp.ansp.br", porém se por acaso o VAX for encaminhador de correio eletrônico do campus, um domínio entre "ifqsc.usp" designando o campus de São Carlos "uspsc" facilitaria para a tabela da FAPESP a administração da correspondência das outras unidades do Campus. Assim denominações "unidade.uspfsc.usp.ansp.br" podem ser todas enviadas para o hospedeiro encaminhador de correspondência do campus, o que diminuiria bastante a tabela do despachador da FAPESP. No entanto isso implica no aumento de um nível de domínio e talvez essa denominação não consiga ser traduzida pelo sistema de domínios de todas as redes. Assim pode-se entender porque nossa denominação mudou tantas vezes e ainda poderá mudar.

Existe aqui no Brasil uma iniciativa da RNP de padronizar os nomes das máquinas. Usando uma notação misturada Internet-X.400, onde os domínios internet são os atributos do X.400, os nomes de usuário têm que ser da forma:

user@ou...ou.o.prdm.br

onde

ou = organization unit

o = organization

prdm = private domain

e C = country , já está preenchido com BR de Brasil.

Um exemplo para um usuário na nossa máquina seria:

user@nome-da-máquina.ifqsc.usp.br

onde

ou = nome-da-máquina

o = ifqsc (também pode ser o = vazio, ou = ifqsc)

prdm = usp

C = br

Considerações Finais

Foi visto neste capítulo que num ambiente de interconexão de redes, onde é necessário tratar a heterogeneidade das várias redes interligadas, os conceitos nome, endereço e rota, bem como máquina retransmissora e despachador de correio são fundamentais. O entendimento desses conceitos permitiu abordar a implementação do Sistema de Correio Eletrônico de forma sistemática. Essa abordagem é apresentada no capítulo 3. O conceito de domínio é utilizado no capítulo 3 para estabelecer limites no ambiente heterogêneo em que o sistema é implementado. A partir dessa identificação de domínios é utilizado, para denominação de recurso, o sistema de domínios internet, como consta na descrição efetuada no capítulo 6.

Capítulo 3

Estruturação de um sistema de Correio Eletrônico

Considerações Iniciais

Para tratar de forma sistemática a implantação do Sistema de Correio Eletrônico são identificadas neste capítulo as diferentes etapas envolvidas na postagem de uma mensagem, associando-as aos vários softwares que as efetivam nos diversos equipamentos. Na sessão 3.1 é feita a identificação das etapas, na sessão 3.2 é tratada a etapa de transmissão por meio físico, na sessão 3.3 a etapa de Interface de Usuário e a sessão 3.4 refere-se às demais etapas.

3.1 Identificação de Etapas

Para a implementação do Sistema de Correio Eletrônico (S.C.E.) local depara-se com o problema da heterogeneidade dos equipamentos. Conforme a fig.3.1 pode-se ver o ambiente no qual o S.C.E. deve ser implementado. Tem-se várias estações de trabalho (Sun, Intergraph) rodando diferentes tipos (BSD 4.2[Sun90b], System V[Cof88]) do sistema operacional UNIX, cada uma com sua própria arquitetura de software e hardware de rede conforme o fabricante. Tem-se também computadores VAX-Digital com sistema operacional VMS e microcomputadores tipo IBM-PC com sistema operacional DOS. Esses equipamentos precisam estar interligados de forma que uma mensagem possa trafegar entre eles.

Referindo-se a domínio como conjunto e subconjunto, tem-se na fig.3.1 limitados por curvas fechadas os domínios IFQSC (Instituto de Física e Química de São Carlos), FPSP (FAPESP), ICMSC (Instituto de Ciências Matemáticas de São Carlos), EMBRAPA (Empresa Brasileira de Pesquisa Agrícola). Computadores dentro de cada domínio terão que ter denominações diferentes conforme o domínio ao qual pertencem.

Para conseguir tratar essa heterogeneidade é preciso identificar o que envolve correio eletrônico. O que acontece quando um usuário deseja enviar uma mensagem via C.E.? A fig.3.2 ilustra o processo.

O usuário interage com um programa que o ajuda a compor a mensagem e denominar o destinatário. Esse programa é identificado como interface com usuário. O computador sendo um retransmissor de correio terá um despachador de correio, que, conforme visto

Figura 3.1: Ambiente do Sistema Implementado

Figura 3.2: O que está envolvido no envio de uma mensagem via C.E.

no capítulo anterior, transformará a denominação do destinatário dada pelo usuário de forma a conseguir encaminhar a mensagem para a rede do entroncamento adequado. Após a transformação da denominação a mensagem deve ser direcionada à rede correta e através de aplicação dos protocolos adequados à rede, a mensagem conseguirá ser transmitida por algum meio físico e atingir o computador destino. Por um caminho inverso a mensagem conseguirá ser lida pelo usuário ao qual é destinada.

Assim pode-se identificar etapas no envio de uma mensagem: uma etapa de interfaceamento com o usuário, uma etapa de tradução nome-destinatário e determinação do mecanismo de transporte, uma etapa de aplicação de protocolos conforme o mecanismo de transporte, e a etapa de transmissão por meio físico.

Assim, após essa estruturação em etapas, pode-se identificar nos vários conjuntos de equipamentos, o que de software e hardware cada etapa dessa envolve (fig.3.3).

Outras opções também foram identificadas e testadas, mas descartadas para o sistema implementado. Exemplo (fig.3.4).

No caso das Interpro's só se conseguia enviar mensagem via UUCP em cima de XNS sobre Ethernet [Int90]. Isso devido a arquitetura de rede do fabricante, que quando a Ethernet está ativa este é o meio de comunicação usado. Assim UUCP quando Ethernet ativo não podia usar o meio serial. Como nenhuma outra estação conversa XNS, as duas Interpro's ficavam isoladas, podendo apenas mandar mensagem uma para outra através do processo ilustrado na fig.3.4a.

O UUCP das Suns permite usar Ethernet através do TCP/IP e serial simultaneamente. Porém, apesar de tanto as Suns como Interpro's poderem usar UUCP em cima de Ethernet, não podiam trocar mensagens pois usavam conjunto de protocolos diferentes, XNS e TCP/IP, em termos do modelo OSI, protocolos a nível da camada de transporte [Tra90].

Verificou-se que o software Sendmail (nas SUN's), além de despachador (só posteriormente identificado) implementa o protocolo SMTP, da camada de aplicação do TCP/IP. Portanto, faz parte do TCP/IP. As Interpro's não vieram com o Sendmail. Só após essa identificação é que nos foi possível solicitar o envio.

Como a interface com o usuário assumida pelo sistema é diferente em cada tipo de estação, adotou-se o software público ELM nos ambientes UNIX para ter o máximo possível de interfaces comuns, (ideal que fossem iguais para todos os equipamentos).

No caso ideal, segundo as opções oferecidas por cada tipo de estação poderia-se ter como despachador de correspondência o Sendmail utilizando como mecanismo de transporte UUCP e TCP/IP; o UUCP por sua vez poderia usar Ethernet, tanto em cima do XNS quanto do TCP/IP, como também serial; o TCP/IP por sua vez usando Ethernet ou serial via SLIP (software público para IP em cima de serial).

No entanto o ideal não é comum para os dois tipos de estação. Assim, o que existe em comum para os dois tipos foi a opção para "estações" no sistema implementado, ilustrado na fig.3.3.

No caso do VAX teria-se para cada etapa:

- A interface de MAIL do VMS
- O despachador de correspondência PMDF, software adquirido da INOSOFT
- Os conjuntos de protocolos:

Figura 3.3: Software e Hardware nas várias etapas de C.E.

Figura 3.4: Opções descartadas

- UUCP software livre fornecido pela DECUS (grupo de usuários digital, que fornece software desenvolvido pelos próprios usuários)
- o software FUSION adquirido, que implementa TCP/IP para VMS compartilhando Ethernet com o DECNET (uma alternativa para o FUSION era o software MULTINET, na época não comercializado para o Brasil, que ao invés de compartilhar com o Decnet torna-se um objeto do Decnet)
- o DECNET que vem com o equipamento e é o software que segue a arquitetura de rede do fabricante

- e para meios físicos as opções: serial, ethernet e X.25 através do PSI.

No caso dos micros têm-se duas opções. A primeira usando UUCP, software público que implementa UUCP para PC, o pacote possui interface com usuário, MAIL-UUCP, e usa a interface serial para comunicação. A segunda usando o KA9Q, software público que implementa TCP/IP para micros, o pacote também possui interface com o usuário, BM, e interfaceia com Ethernet ou serial para comunicação.

3.2 Transmissão por meio físico

Na etapa transmissão por meio físico pode-se identificar três ocorrências comuns: serial, Ethernet e X.25.

Serial

A interface serial é encontrada em todos os equipamentos (aqui relacionados). Em geral a interface utiliza o padrão RS232-C, encarado dentro do modelo OSI, protocolo da camada física [Tra90], e para meio de transmissão o par-trançado [Tra90].

Ethernet

A interface Ethernet é encontrada na maioria dos equipamentos de maior porte. Para micros existem placas (ex: NE1000) que se adquiridas permitem PCs possuírem interface

Ethernet. Existe o conceito de “packet driver” nos softwares de domínio público. Esse conceito permite que softwares públicos sejam escritos de modo a interfacear com um “packet driver”. E também públicos existem “packet driver’s” escritos para várias placas que emulam Ethernet. Isso permite aos softwares não se prenderem muito aos detalhes de cada fabricante, deixando aos “packet driver’s” esta incumbência. Também permite que uma placa de determinado fabricante que possua um “packet driver” público, escrito para ela, seja usada com qualquer software que interfaceia com “packet driver”. Na implementação utilizou-se o *packet driver* para NE1000 obtido do sun.soe.clarkson.edu.

O padrão Ethernet é um projeto conjunto da Digital, Xerox e Intel [MB76] [S⁺82] [Con83]. Em termos do modelo OSI inclui a camada física e de enlace de dados [Lam84] [Wei83] [Sta84]. Utiliza topologia de barramento [Tra90], cabo coaxial blindado, possui taxa de transmissão de 10Mb/s, usando multiacesso distribuído ao canal de comunicação e o controle de acesso CSMA/CD [Lam80] [DEC82] [Lam83] [Gui89]. Possui as opções THIN e THICK que alteram o comprimento máximo de um segmento, máximo número de nós, mínimo espaçamento entre os nós e tipos de conectores. O artigo “Ethernet: A Unix LAN” de Charles Spargon de [Wai87] dá uma visão bem prática das conexões Ethernet.

X.25

A recomendação X.25 do CCITT e relacionados [Bur83] [DK86] [Uns82] [Ryb80] é utilizado para acesso a computadores conectados a rede de comutação de pacotes brasileira RENPAC [EMBa] [EMBb] [EMBe] [EMBc] [EMBf] [EMBd] [EMBg] [Rob78] [CK78]. O acesso atual do VAX local com a FAPESP é via RENPAC. A recomendação [Tra90], segundo o modelo OSI envolve até a camada de rede. Equipamentos operando no modo síncrono (pacote), deverão implementar o protocolo X.25. Para equipamentos operando de forma assíncrona, a rede (RENPAC) possui interfaces PAD (Packet Assembly Disassembly), cujas especificações estão contidas nas recomendações X.3, X.28 e X.29. Os endereços nas redes públicas são definidos pela recomendação CCITT X.121 que pretende um plano de numeração internacional. A cada rede pública de comutação de pacote é atribuído um código de identificação de dados (DNIC). Um DNIC é um número de quatro dígitos, onde os primeiros três dígitos identificam o país e o quarto dígito identifica a particular rede pública de comutação de pacotes dentro do país. Por exemplo, o DNIC da TELENET (EUA) é 3110, enquanto que o DATAPAC (França) é 3020 e o RENPAC é 0724. Segue-se ao DNIC, um endereço internacional que contém um número do terminal de rede (NTN) que especifica o endereço do terminal dentro da rede pública de dados. O NTN é um número de 1 a 10 dígitos e é atribuído a um particular terminal pela administração da rede pública de dados. Geralmente o encaminhamento das chamadas entre redes públicas de dados é baseado nos DNIC's, enquanto o encaminhamento das chamadas dentro de uma rede pública é baseado nos NTN's.

Como exemplo, o endereço do VAX do IFQSC dentro da RENPAC é 11620020, e do VAX da FAPESP é 11182594. Para acesso de alguma rede pública no exterior o nosso endereço seria **072411620020**. O endereço de um computador na Califórnia (EUA) seria **31102130021999** (CALTECH).

3.3 Interface de usuário

Na etapa de interface com usuário identificamos quatro ocorrências: MAIL-VMS, ELM, MAIL-UUPC e BM-KA9Q.

A seguir são apresentadas algumas sessões de correio, utilizando as interfaces: VMS-mail [Dig88a], ELM, /bin/mail e /usr/ucb/mail [Sun90b]

Sessão com /bin/mail na floyd para enviar mensagem:

```
floyd% /bin/mail gonzalo@uspfs
note que nao pediu o assunto

la no uspfsc observe a montagem do cabecalho
o to: alem do subject: tambem nao aparece
^D
floyd%
```

Sessão com VMS-mail no USPFSC para receber mensagem:

```
New mail on node USPFSC from IN%"monica@floyd.ifqsc.usp.ansp.br"
$ mail
```

You have 1 new message.

MAIL> <return>

```

      #1                27-JUN-1991 02:03:01.36                NEWMAIL
From:   IN%"monica@floyd.ifqsc.usp.ansp.br"
To:
CC:
Subj:

Received: from floyd.ifqsc.usp.ansp.br by USPFSC.IFQSC.USP.ANSP.BR; Thu, 27 Jun
  91 02:02 BRT
Received: by floyd.ifqsc.usp.ansp.br (4.1/SMI-4.1) id AA01751; Thu, 27 Jun 91
  02:05:56-040
Date: Thu, 27 Jun 91 02:05:56-040
From: monica@floyd.ifqsc.usp.ansp.br
Message-id: <9106270605.AA01751@floyd.ifqsc.usp.ansp.br>
Apparently-To: gonzalo@uspfs
X-Envelope-to: gonzalo
```

note que nao pediu o assunto

```
la no uspfsc observe a montagem do cabecalho
o to: alem to subject: nao aparece
```

MAIL>

Sessão no USPFSC com VMS-mail para responder mensagem:

```
MAIL> re
To:     IN%"monica@floyd.ifqsc.usp.ansp.br"
```

Your mail will be delivered by PMDF V3.1

Subj: note que pediu o assunto, pois nao tinha na original

volta para floyd
e leia com /bin/mail

note la na floyd que o /bin/mail mostra a ultima mensagem
e nao um cabecalho de mensagens

MAIL>

Sessão na floyd com /bin/mail para receber mensagens:

```
floyd% /bin/mail
From GONZALO@USPFSC.ifqsc.usp.ansp.br Thu Jun 27 02:22:34 1991
Return-Path: <GONZALO@USPFSC.ifqsc.usp.ansp.br>
Received: from USPFSC.IFQSC.USP.ANSP.BR by floyd.ifqsc.usp.ansp.br (4.1/SMI-4.1
)
        id AA01773; Thu, 27 Jun 91 02:22:33-040
Date: Thu, 27 Jun 91 02:18 BRT
From: Gonzalo Travieso <GONZALO@USPFSC.ifqsc.usp.ansp.br>
Subject: note que pediu o assunto, pois nao tinha na original
To: monica@floyd.ifqsc.usp.ansp.br
Message-Id: <54525132137F00053B@USPFSC.IFQSC.USP.ANSP.BR>
X-Envelope-To: monica@floyd.ifqsc.usp.ansp.br
X-Vms-To: IN%"monica@floyd.ifqsc.usp.ansp.br"
X-Vms-Cc: GONZALO
```

volta para floyd
e leia com /bin/mail

note la na floyd que o /bin/mail mostra a ultima mensagem
e nao um cabecalho de mensagens

Sessão com /usr/ucb/mail na floyd para enviar mensagem:

```
floyd% /usr/ucb/mail gonzalo@uspfsfsc.hepnet
Subject: note que este ja pede o assunto
```

e monta o to: corretamente quando recebe no vax
~D
floyd%

Sessão com VMS-mail no USPFSC para receber mensagem

```
New mail on node USPFSC from IN%"monica@floyd.ifqsc.usp.ansp.br"
#1          27-JUN-1991 02:42:40.33          NEWMAIL
From:      IN%"monica@floyd.ifqsc.usp.ansp.br"
To:        gonzalo@USPFSC.HEPNET
CC:
Subj:      note que este ja pede o assunto

Received: from floyd.ifqsc.usp.ansp.br by USPFSC.IFQSC.USP.ANSP.BR; Thu, 27 Jun
91 02:42 BRT
Received: by floyd.ifqsc.usp.ansp.br (4.1/SMI-4.1) id AA01797; Thu, 27 Jun 91
```


02:46:31-040
Date: Thu, 27 Jun 91 02:46:31-040
From: monica@floyd.ifqsc.usp.ansp.br
Subject: note que este ja pede o assunto
To: gonzalo@USPFSC.HEPNET
Message-id: <9106270646.AA01797@floyd.ifqsc.usp.ansp.br>
X-Envelope-to: gonzalo@USPFSC.HEPNET

e monta o to: corretamente quando recebe no vax

MAIL>

Sessão com VMS-mail no USPFSC para responder mensagens:

MAIL>re

Subj: RE: note que este ja pede o assunto

Note que como a mensagem veio com o assunto,
o assunto e' montado automaticamente como resposta do anterior
la na floyd, quando for lido pelo /usr/ucb/mail
aparecera a lista de mensagens a serem lidas

MAIL>

Sessão com /usr/ucb/mail na floyd para receber mensagens

/usr/ucb/mail

Mail version SMI 4.0 Wed Feb 7 23:50:57 PST 1990 Type ? for help.

"/usr/spool/mail/monica": 2 messages 2 new

>N 1 GONZALO@USPFSC.ifqsc.usp.ansp.br Thu Jun 27 02:22 19/709 note que ped
iu o assunto,

N 2 GONZALO@USPFSC.ifqsc.usp.ansp.br Thu Jun 27 02:55 17/681 Re: note que
este ja pede

Mail> 2

Message 2:

From GONZALO@USPFSC.ifqsc.usp.ansp.br Thu Jun 27 02:55:02 1991

Date: Thu, 27 Jun 91 02:50 BRT

From: Gonzalo Travieso <GONZALO@USPFSC.ifqsc.usp.ansp.br>

Subject: Re: note que este ja pede o assunto.

To: monica@floyd.ifqsc.usp.ansp.br

Note que como a mensagem veio com o assunto,
o assunto e montado automaticamente como resposta do anterior
la na floyd, quando for lido pelo /usr/ucb/mail
aparecera a lista de mensagens a serem lidas

Mail> q

Saved 2 messages in /home2/floyd/monica/Mail/mbox
floyd%

Sessão com o ELM na floyd para enviar mensagem:

floyd% elm

* nova tela

Mailbox is '/usr/spool/mail/monica' with 4 messages [ELM 2.2 PLO] N 1 Jun 27
 Gonzalo Travieso (15) mensagem mais recente
 N 2 Jun 27 Gonzalo Travieso (15) outra mensagem
 3 Jun 27 Gonzalo Travieso (16) outra mensagem
 4 Jun 27 Gonzalo Travieso (16) aparece no cab. do elm

You can use any of the following commands by pressing the first character;
 d)delete or u)ndelete mail, m)ail a message, r)eply or f)orward mail, q)uit
 To read a message, press <return>. j = move down, k = move up, ? = help

Mail sent! Command: Mail

Send the message to: gonzalo@uspfc

To: gonzalo@uspfc

Subject of message: com elm

Copies to:

Invoking editor... Please choose one of the following options by parenthesized
 letter: se)dit message, edit h)eaders, s)end it, or f)orget it

Sessão com VMS-mail no USPFSC para receber mensagem:

#1 27-JUN-1991 03:17:41.18 NEWMAIL
 From: IN%"monica@floyd.ifqsc.usp.ansp.br"
 To: gonzalo@USPFSC.IFQSC.USP.ANSP.BR
 CC:
 Subj: com elm

Received: from floyd.ifqsc.usp.ansp.br by USPFSC.IFQSC.USP.ANSP.BR; Thu, 27 Jun 91 03:17 BRT

Received: by floyd.ifqsc.usp.ansp.br (4.1/SMI-4.1) id AA01852; Thu, 27 Jun 91 03:21:30-040

Date: Thu, 27 Jun 91 3:21:29 AST

From: monica@floyd.ifqsc.usp.ansp.br

Subject: com elm

To: gonzalo@USPFSC.IFQSC.USP.ANSP.BR

Message-id: <9106270721.AA01852@floyd.ifqsc.usp.ansp.br>

X-Mailer: ELM [version 2.2 PLO]

X-Envelope-to: gonzalo

Esse e bem mais interativo,

Press RETURN for more...

MAIL>

#1 27-JUN-1991 03:17:41.18 NEWMAIL
 tambem, com isso, exigente no terminal.
 Os varios estados por que passa a mensagem e'
 mostrado no campo Comand:. Note o ultimo: MailSent!
 Na hora do envio aparece comandos para: enviar,
 esquecer, etc...

MAIL>

Sessão com ELM na floyd para receber mensagens:

```
floyd% elm
* nova tela
Mailbox is '/usr/spool/mail/monica' with 4 messages [ELM 2.2 PLO] N 1 Jun 27
Gonzalo Travieso (15) mensagem mais recente
  N 2 Jun 27 Gonzalo Travieso (15) outra mensagem
    3 Jun 27 Gonzalo Travieso (16) outra mensagem
    4 Jun 27 Gonzalo Travieso (16) aparece no cab. do elm
```

You can use any of the following commands by pressing the first character;
d)delete or u)ndelete mail, m)ail a message, r)eply or f)orward mail, q)uit
To read a message, press <return>. j = move down, k = move up, ? = help
Command:<return>

```
* nova tela
Message 3/4 From Gonzalo Travieso Jun 27 '91 at 3:26 a
m
```

```
Return-Path: <GONZALO@USPFSC.ifqsc.usp.ansp.br>
Date: Thu, 27 Jun 91 03:26 BRT
Subject: outra mensagem
To: monica@floyd.ifqsc.usp.ansp.br
X-Envelope-To: monica@floyd.ifqsc.usp.ansp.br
X-Vms-To: IN%"monica@floyd"
X-Vms-Cc: GONZALO
```

oi

Command ('i' to return to index): i

* o 'i' volta a mostrar a tela anterior

O artigo "ALL About UNIX Mailers" de Dave Taylor em [Wai87] apresenta um excelente estudo comparativo entre várias interfaces de usuário em ambiente Unix. A opção ELM para o sistema Unix, por enquanto, é a mais adequada.

O padrão X.400 dá diretrizes, tabela 3.1, para os serviços contidos em uma interface de usuário.

3.4 Demais Etapas

Na etapa aplicação de protocolos, segundo o mecanismo de transporte, pode-se identificar três ocorrências principais, não considerando os softwares distintos que as implementam: UUCP, DECNET e TCP/IP. TCP/IP usando como meio de transmissão a interface serial não é comum nas implementações dos vários fabricantes. No entanto, existem softwares públicos seguindo a especificação RFC1055 [Rom90] [Jac90] que implementam IP em cima de serial (SLIP). O KA9Q possui essa implementação embutida. No sistema implementado está-se usando SLIP para as Sun's conseguido na base de software SIMTEL20. O DECNET permite interface serial, porém esse fato não foi usado

Tabela 3.1: Diretrizes para serviços em X.400

Serviço	Objetivo obs: AU - Agente de usuário
Indicação do Remetente	Nome do Remetente amigável para o usuário
Designação de usuários autorizados	Para assinaturas não legais
Indicação de Destinatários Primários & Com Cópia	PARA: e CC:
Destinatários com cópia oculta	CCO:
Indicação do Assunto	Assunto do <i>memorandum</i>
Identificação da Mensagem	Identificador não ambíguo e único (no contexto do AU remetente) que os usuários podem utilizar como referência
Referência Cruzada	Inclue os identificadores de mensagem com que a presente está relacionada
Mensagens Obsoletas	Inclue os identificadores de mensagem que a presente deixa sem efeito
Data de expiração	Após essa data o remetente considera a mensagem inválida
Sensibilidade	Especifica o grau de intimidade da mensagem: Pessoal, Privada, Confidencial da Empresa
Importância	Indica o grau de urgência com que requer a atenção do destinatário. Pode ser: pouca, normal ou elevada
Solicitação da Resposta	Para quem e quando
Indicação da Resposta	Identificador da Mensagem a que se contesta
Notificação de Recebimento	Conta ao remetente que o destinatário tomou alguma ação sobre sua mensagem
Notificação de Não Recebimento	Conta ao remetente que o destinatário não retorna recebimento (ex: por segurança pessoal)
Indicação de Auto-reenvio	Indica que o AU destinatário reenviou a mensagem automaticamente. Por exemplo, um usuário em férias pode instruir seu AU para que reexpida as mensagens para seu substituto

para o sistema implementado, e portanto não será tratado aqui.

Na etapa de tradução nome-destinatário e determinação do mecanismo de transporte, pode-se identificar duas ocorrências principais: PMDF e Sendmail.

Considerações Finais

A partir das etapas identificadas neste capítulo e as suas respectivas ocorrências principais são detalhadas nos próximos capítulos as etapas não tratadas aqui. A etapa de tradução de denominação e determinação do mecanismo de transporte é assunto do capítulo 4 onde são descritas as principais ocorrências dessa etapa: PMDF e Sendmail. A etapa de protocolagem com as ocorrências TCP/IP, DECNET e UUCP é assunto do capítulo 5. Os diferentes softwares que implementam as ocorrências de cada etapa serão detalhados no capítulo 6, pois através das configurações destes se efetiva o sistema implementado.

A fig.3.5 ilustra a interligação dos vários equipamentos, identificados os softwares e hardwares envolvidos nas primeira e segunda etapas de correio eletrônico.

Figura 3.5: Equipamentos interligados com a identificação do Software e Hardware envolvidos na primeira e segunda etapas de C.E.

Capítulo 4

Tradução de Denominações e Determinação do Mecanismo de Transporte

Considerações Iniciais

Como foi visto no capítulo 3 entre as etapas envolvidas na estruturação de um sistema de correio eletrônico, figura a etapa de tradução de denominação de determinação do mecanismo de transporte. Os despachadores que efetivam essa etapa no sistema implementado são o PMDF, analisado em 4.1, e o Sendmail, analisado em 4.2.

O NIS e SMTP serão introduzidos aqui apenas para entender as referências feitas a eles nesse capítulo.

Máquinas Sun rodando NIS (Network Information Service) [Sun90a] permitem ao administrador de um sistema de máquinas interligadas, distribuir base de dados administrativas (mapas) por várias máquinas. Permite atualizar essas bases a partir de um local central de maneira automática e confiável, assegurando desse modo que todos os clientes compartilhem as mesmas bases de modo consistente através da rede. Entre essas bases estão o arquivo de usuários autorizados do sistema, *passwd*, o arquivo de nomes de hospedeiros e endereços internet usado no TCP/IP, *hosts*, e o arquivo de sinônimos, *aliases*.

SMTP ([Com88], [Pos82], [Tra90]) é um protocolo interativo que define uma transação entre dois servidores de transporte de correspondência. Um é definido como o que envia (S) e o outro como o que recebe (R).

Um exemplo de uma transação SMTP seria:

S: ALO CUVMB.BITNET

R: 250 CUVMA.BITNET

S: MAIL FROM:<EACUS@CUVMB.BITNET>

R: 250 OK

S: RCPT TO:<AC2US@CUVMA.BITNET>

R: 250 OK

S: RCPT TO:<FOO@CUVMA.BITNET>

R: 551 No such user

S: DATA

R: Start mail input, end with record containing only "."

Date: Monday, 13-Sep-82 10:32 EST

From: Alan Crosswell <EACUS@CUVMB>

Subject: Hello

To: AC2US@CUVMA

Hello there ...

4.1 Descrição do despachador PMDF

PMDF (Pascal Memo Distribution Facility) [FV89] é um sistema de propósito geral para entrega de correspondência baseada em computador. PMDF provê um ambiente de distribuição uniforme que pode interfacear com várias interfaces de usuário e vários mecanismos de transporte. A versão que se está usando nessa implementação, chamada PMDF-822 V.3.1, usa o MAIL-VMS como interface de usuário. Isto é, toda interação usuário (não administrador) e PMDF é feita via MAIL-VMS.

Mensagens a serem tratadas pelo PMDF devem ser "endereçadas" da forma:

IN%"denominação-internet" via MAIL-VMS. Ou se o software está rodando num entroncamento remoto com o qual se possui ligação Decnet, devem ser "endereçadas" da forma:

entroncamento::IN%"denominação-internet" via MAIL-VMS.

O padrão RFC-822, que o PMDF segue, especifica um grande número de campos no cabeçalho. Infelizmente, o MAIL-VMS provê apenas os campos "From:", "To:" e "Subject:" como informação de cabeçalho. Esse fato deixa ao PMDF duas opções: ou descarta qualquer outro campo ou o torna parte do corpo da mensagem. Nenhuma opção é completamente satisfatória: eliminar cabeçalhos provoca perda de informação valiosa, porém a inserção de cabeçalhos na mensagem interfere com a operação correta do comando de extração de cabeçalho do MAIL-VMS.

Campos adicionais podem ser fornecidos pelo usuário através do estabelecimento de nomes lógicos (via operação normal de usuário com o VMS) tais como: PMDF-comments, PMDF-errors-to, PMDF-from, PMDF-keywords, PMDF-references, PMDF-reply-to, PMDF-timezone e PMDF-x-organization.

A informação que realmente aparece no cabeçalho "From:" não é necessariamente o MAIL-VMS "From:", pois o PMDF tenta usar para o "From:" o campo que contém a

denominação mais provável para o sucesso de uma réplica. Os campos usados na ordem de prioridade são:

- (1) Resent-Reply-to: (se presente)
- (2) Resent-From: (se presente)
- (3) Reply-to: (se presente)
- (4) From: (se presente)
- (5) Resent-Sender: (se presente)
- (6) Sender: (se presente)
- (7) endereço de retorno de envelope (sempre presente)

O termo envelope, se refere ao que é acrescentado à mensagem para efeito de transporte. A linha de cabeçalho "X- Envelope-to:" não é estabelecida pelo usuário. Esta linha contém uma duplicata da lista de denominação-destino que aparece no envelope da mensagem. Reflete o envelope destino mais recente, no caso de repasse da mensagem. É útil para o PMDF reconstruir endereços de envelope danificados por vários mecanismos de transporte (principalmente a BITNET que os trunca em 8 caracteres). É útil também para depuração pois esse campo, numa mensagem que retorna com erro, contém o endereço de envelope justamente antes do erro.

Canal

O principal conceito no PMDF é o canal. Um canal é alguma forma de conexão com outro sistema de computador ou grupo de sistemas. O hardware da conexão real e o software do mecanismo de transporte pode variar bastante de um canal para outro. Cada canal consiste de um ou mais programas-de-canal e uma fila que armazena as mensagens a serem enviadas para um ou mais sistemas associados ao canal. Os programas-de-canal são partes do PMDF que interfaceiam com os vários mecanismos de transporte. Os programas-de-canal executam duas funções: (1) transmitem mensagens a sistemas remotos, eliminando-as da fila associada ao canal, após serem enviadas, e (2) aceitam mensagens de sistemas remotos, colocando-as na fila associada ao canal.

Um programa-de-canal pode ser "master" ou "slave" conforme inicia a transferência a um sistema remoto ou aceita a transferência iniciada por um sistema remoto.

Os canais no PMDF possuem nomes com 12 caracteres. Apenas letras, números, e os sinais de "-" e "\$" podem ser usados. Letras maiúsculas e minúsculas são usadas indistintamente. Alguns nomes de canais são reservados para uso particular. O canal L é o canal local, usado para entregar correspondência a usuários do sistema local e sistemas acessíveis via Decnet Mail ou PSIMail. O canal P é o Phonenet genérico, usado para comunicar com um hospedeiro Phonenet central. O canal D é o Decnet MAIL, usado para entregar mensagens por Decnet via MAIL-VMS. Nomes de canais iniciados com os prefixos da tabela 4.1 são de tipo determinado. Isto implica que programas-de-canal especiais serão invocados para servir esses canais.

Arquivo de configuração do PMDF

Para determinar em qual fila (associada a que canal) uma mensagem deve ser colocada, o PMDF consulta o arquivo de configuração PMDF.CNF. Além disso esse arquivo contém regras de reescrita que convertem denominações curtas para nomes-internet (nomes com

estilo domínio). O arquivo de configuração do PMDF é um arquivo de texto ASCII que pode ser criado ou modificado com qualquer editor de texto. O arquivo consiste de duas partes: regras de reescrita de domínio e tabela canal/hospedeiro. As regras aparecem primeiro no arquivo e são separadas da tabela canal/hospedeiro por uma linha em branco. Linhas de comentário iniciam com ponto de exclamação. À página 50 tem-se ilustrado o aspecto desse arquivo.

Regras de Reescrita para Domínio

Regras de reescrita para domínio são usadas para converter denominações comuns para denominações com domínios verdadeiros e para determinar seus canais correspondentes. Essas regras são usadas para reescrever tanto denominações da camada de transporte quanto denominações das várias linhas de cabeçalho da mensagem. A camada de transporte é o envelope da mensagem, que contém informação para encaminhamento e é invisível ao usuário.

Cada regra aparece numa única linha. Comentários, mas não linhas em branco, são permitidos entre regras. Cada regra consiste de duas partes: um padrão seguido de uma cadeia de equivalência ou máscara. As duas partes devem ser separadas por um ou mais espaços.

Extração da primeira especificação hospedeiro/domínio O processo de reescrita começa pela extração da primeira especificação hospedeiro/domínio. Essa especificação é procurada (percorrida) na denominação dos seguintes modos:

- hospedeiros em encaminhamentos fontes (lido da esquerda para direita)
- hospedeiros que aparecem do lado direito do sinal "@"
- hospedeiros que aparecem do lado direito do último sinal de "%"
- hospedeiros que aparecem do lado esquerdo do primeiro ponto de exclamação.

Exemplos:

Denominações	Extração da 1ª especific. hosp./domínio
--------------	---

usuário@a	a
usuário@a.b.c	a.b.c
@a:usuário@b.c.d	a
a!usuário	a
a!usuário@b	b
a!usuário%b	b
a!usuário%b@c	c

O padrão RFC822 não diz nada a respeito da interpretação de pontos de exclamação e sinais de por cento na denominação. Como é costume interpretar sinais de "%" da mesma forma que sinais de "@", se esse não está presente, o PMDF adotou essa convenção.

A interpretação do ponto de exclamação é conforme as convenções RFC976 de denominações ("bangstyle"), e torna possível usar a denominação estilo UUCP no PMDF.

Percorrendo as regras de reescrita Uma vez que a primeira especificação hospedeiro/domínio foi extraída da denominação, o PMDF consulta as regras e compara com a parte do padrão de cada regra. Se não casa com nenhuma regra, a primeira parte da especificação de domínio é retirada (a parte antes do ponto: usualmente o nome do hospedeiro) e outra tentativa é feita (com o subdomínio). O processo continua até casar alguma regra ou esgotar toda a especificação hospedeiro/domínio.

O efeito do método é tentar casar domínios mais específicos primeiro e assim trabalhar do domínio mais específico ao mais geral. Por exemplo, suponha-se a denominação “dan@sc.cs.cmu.edu”. Primeiro serão percorridas as regras com o padrão “sc.cs.cmu.edu”, depois com “cs.cmu.edu”, depois com “.cmu.edu”, depois com “.edu”, e finalmente com o padrão “.”.

A regra com o padrão “.” pode ser usada para tratar toda especificação que não bate com nada. Serve como regra “default” e pode ser usada se a validação da denominação é deixada para outro sistema.

Máscara das regras de reescrita Uma vez que a especificação hospedeiro/domínio casou com o padrão de alguma regra, a denominação é reescrita usando a parte da máscara da regra. A máscara especifica três coisas: (1) um novo nome de usuário para a denominação, (2) uma nova especificação hospedeiro/domínio, e (3) o nome de um sistema anexo a um canal existente do PMDF (“sistema encaminhador”) para o qual as mensagens serão realmente enviadas.

O formato usual da máscara é “a%b@c”, onde “a” é o nome do usuário, “b” é a nova especificação hospedeiro/domínio e “c” é o sistema encaminhador. Se “b” e “c” são idênticos omite-se o “%b”.

Cadeias de substituição são permitidas nas máscaras. Exemplo: toda a ocorrência de “\$U” na máscara será substituída com o nome do usuário da denominação original. As cadeias de substituição são:

\$D porção da especificação de domínio que “casou”

\$H porção da especificação de domínio que não “casou” (à esquerda do ponto do padrão)

\$L parte não casada de um domínio literal (direita do ponto do padrão literal)

\$U nome de usuário da denominação original

\$\$ sinal \$

\$% sinal %

\$@ sinal @

\$(texto) substituição geral de uma base de dados

A forma “\$(texto)” é tratada de maneira especial. O “texto” é usado como chave para acesso a um arquivo de base de dados (GENERAL.DAT). Se o texto é casado na base a máscara correspondente extraída é substituída e a base é percorrida novamente para substituições adicionais. Para evitar referências recursivas indefinidas é proibido a ocorrência “\$(texto)” na máscara extraída.

O formato de máscara “a%b” é usado para meta-regras que requerem reescritas adicionais após sua aplicação. Quando o padrão “a%b” é encontrado, “a” torna-se o novo nome de usuário e “b” torna-se a nova especificação hospedeiro/domínio, e então todo processo de reescrita é repetido na denominação resultante. Por exemplo, a regra

`.removível $U%$H`

tem o efeito de retirar todas as ocorrências “.removível” de domínio do final das denominações.

O formato de máscara “a@b@c” trabalha da mesma forma que o formato usual “a%b@c”, exceto que o sistema encaminhador “c” será também inserido na denominação como um encaminhamento fonte. Essa inclusão pode ser necessária para alguns canais que têm que estabelecer conexão com um sistema encaminhador e determinar o nome do sistema pela denominação “To:” do envelope.

O formato de máscara “a@b@c@d”, usa “c” como encaminhamento fonte e “d” como sistema encaminhador.

Se a especificação hospedeiro/domínio não casa com nenhuma regra, e não há regra “default”, PMDF usa a especificação original para a nova especificação e para sistema encaminhador. Detecta-se uma denominação que não faz sentido quando o sistema encaminhador não casa com nenhum nome de sistema da tabela de canais.

Tabela Canal/Hospedeiro

A segunda parte do arquivo de configuração é a tabela canal/hospedeiro. Essa tabela define os canais que o PMDF pode usar e os nomes dos sistemas associados a cada canal. A tabela consiste de blocos individuais cada qual descrevendo um canal. Blocos são separados por uma linha em branco. Comentários podem aparecer dentro de um bloco de canal, mas não linhas em branco.

O primeiro bloco de canal no arquivo descreve o canal “local”, usado para entregar mensagens ao sistema local.

A primeira linha do bloco de canal possui o nome do canal (até 12 caracteres) e seguido por um espaço várias chaves modificadoras opcionais (tabela 4.2) separadas por espaço. A segunda linha do bloco de canal especifica o nome de hospedeiro oficial associado ao canal, seguido de uma apelido opcional para o hospedeiro local. A linha é da forma:

nome-de-hospedeiro-oficial apelido-hospedeiro-local

O nome-de-hospedeiro-oficial seria o nome completo (incluindo domínios e sub-domínios) do hospedeiro com o qual o canal se comunica.

O nome da máquina hospedeira local é normalmente conhecida pelo nome-de-hospedeiro-oficial do primeiro bloco de canal (canal “1”) no arquivo de configuração. Contudo, algumas vezes seria útil que o hospedeiro-local tivesse nomes diferentes dependendo do canal sendo usado. Essa situação normalmente surge quando uma máquina é conectada a mais de uma rede. Por exemplo, um sistema pode querer ser conhecido como “ymir.uucp” na rede UUCP, “ymir.claremont.edu” na Internet e “ymir.bitnet” na BITNET.

O apelido do hospedeiro local na segunda linha do bloco de canal provê essa funcionalidade. Se esse apelido está especificado, ele é usado ao invés do nome normal

do hospedeiro local nos endereços associados ao canal. Assim cada canal pode ter seu próprio apelido para o hospedeiro local.

Qualquer linha adicional no bloco de canal especifica hospedeiros adicionais ou apelidos para hospedeiros que o canal pode alcançar. Essas linhas são da forma:

nome-hospedeiro nome-conveniente

Mensagens para o sistema “nome-hospedeiro” serão enfileiradas nesse canal, mas as denominações “To:” serão reescritas na camada de transporte (envelope) para o “nome- conveniente” ao invés do “nome-hospedeiro”. No cabeçalho da mensagem as denominações não serão reescritas. Se o “nome- conveniente” é omitido o nome de hospedeiro oficial do canal será usado, assim o “nome-hospedeiro” é portanto um apelido para o hospedeiro oficial.

Casando canais com denominações

As regras de reescrita e a tabela de canal cooperam afim de determinar a disposição de cada denominação. O resultado do processo de reescrita é uma denominação reescrita e um “sistema encaminhador”, isto é, o sistema para o qual a mensagem é para ser enviada. O sistema encaminhador é apenas o primeiro passo ao longo do percurso que a mensagem toma para atingir seu destino. De fato, o sistema encaminhador frequentemente não aparece em qualquer forma de denominação real da mensagem.

Após o término do processo de reescrita é feita a busca do sistema encaminhador na tabela de canal. Se o sistema encaminhador não é encontrado, a denominação está com erro. Se o sistema encaminhador é encontrado, o canal que serve o sistema encaminhador é o canal onde a mensagem será enfileirada.

Base de dados de denominação reversa

A denominação do cabeçalho “From:” recebe um tipo adicional de processamento. O PMDF usa tudo, usuário@hospedeiro mais encaminhamentos fontes, como chave de acesso a um arquivo especial REVERSE.DAT. Se a denominação é encontrada, é substituída pela máscara correspondente obtida na base. Essa facilidade especial é ativada automaticamente se existe a base REVERSE.DAT. O assumido é a não existência da base.

O uso primário de uma base reversa é a substituição de uma denominação genérica (talvez uma denominação de máquina central) para denominações de sistemas remotos e possivelmente transitórios ou vice-versa. Essa é a única facilidade de reescrita do PMDF que se aplica à denominação como um todo; todas as outras facilidades de reescrita do PMDF se aplicam a partes da denominação.

Arquivo para Apelidos

O PMDF possui a facilidade de prover nomes de caixa postal no canal local que não correspondem a usuários reais do sistema local. Tais apelidos são úteis para construção de listas de postagem, repasse de mensagens e apelidos para nomes de usuários.

Os apelidos são guardados num arquivo central ALIASES. . Cada vez que uma denominação para um canal local é encontrada, a caixa-postal (ou nome de usuário) especificada na denominação é comparada com cada entrada do arquivo. Se ocorre algum

casamento (equivalência) a caixa-postal é substituída pelo valor ou valores especificados no arquivo. Os valores produzidos pelo apelido são tratados como denominações completamente novas e reanalisadas do zero.

O formato do arquivo ALIASES. é:

apelido1: x1,...,xM

.

apelidoN:

onde apelido1 é traduzido para x1,...,xM. Cada valor x pode ter até 80 caracteres e cada apelido até 40 caracteres. Linhas podem continuar com “,” seguido de “\”. Linhas de comentário iniciam com “!”.

Exemplo do arquivo de Configuracao do PMDF

```

!*** Regras de Reescrita (nao existem linhas em branco)
!
! Esse primeiro conjunto de regras nao serve para contexto real,
! apenas ilustra e enfatiza a determinacao do sistema encaminhador
! e o enfileiramento no canal associado.
a      $U@a
b      $U@b
c      $U%c@b
d      $U%d@a
! Assim essas regras e as definicoes do canais a_canal e b_canal
! farao com que as denominacoes mostradas abaixo sejam
! encaminhadas e enfileiradas do seguinte modo:
!
!   denominacoes      encaminhadas a      enfileiradas no canal
!   -----
!   u@a                a                    a_canal
!   u@b                b                    b_canal
!   u@c                b                    b_canal
!   u@d                a                    a_canal
!   u@e                e                    a_canal
!   u@f                f                    b_canal
!
! Esse conjunto de regras provoca que todas as denominacoes
! que possuem "hospedeiro.dominio" (nome do hospedeiro local)
! ou o seu apelido "hospedeiro", sejam encaminhadas ao canal local
! e as denominacoes reescritas da forma completa "hospedeiro.dominio"
hospedeiro      $U@hospedeiro.dominio
hospedeiro.dominio  $U@hospedeiro.dominio
!
! Esse conjunto de regras transforma nomes curtos ou apelidos
! para a forma canonica (nomes estilo dominio). Alem disso

```

```

! encaminha para sistemas associados ao canal d (decnet)
CLARMATH.CLAREMONT.EDU      $U@CLARMATH.CLAREMONT.EDU
CLARMATH                    $U@CLARMATH.CLAREMONT.EDU
CMATH                      $U@CLARMATH.CLAREMONT.EDU
CMCVAX.CLAREMONT.EDU      $U%CMCVAX.CLAREMONT.EDU@DECNET-MAIL
CMCVAX                    $U%CMCVAX.CLAREMONT.EDU@DECNET-MAIL
ECHMC.CLAREMONT.EDU      $U%ECHMC.CLAREMONT.EDU@DECNET-MAIL
ECHMC                    $U%ECHMC.CLAREMONT.EDU@DECNET-MAIL
MEDCHM.CLAREMONT.EDU      $U@MEDCHEM.CLAREMONT.EDU
MEDCHM                    $U@MEDCHEM.CLAREMONT.EDU
MEDCHEM.CLAREMONT.EDU      $U@MEDCHEM.CLAREMONT.EDU
MEDCHEM                    $U@MEDCHEM.CLAREMONT.EDU
!
! Esse conjunto de regras assume que todas as denominacoes
! terminam com um dos dominios [.ARPA,.BITNET,...] e que o
! acesso a esses dominios 'e feito pelo sistema retransmissor
! "relay.cs.net" atraves de um canal phonenet.
.ARPA                      $U%$H.ARPA@relay.cs.net
.BITNET                    $U%$H.BITNET@relay.cs.net
.MAILNET                   $U%$H.MAILNET@relay.cs.net
.CSNET                     $U%$H.CSNET@relay.cs.net
.EDU                       $U%$H.EDU@relay.cs.net
.COM                       $U%$H.COM@relay.cs.net
.MIL                       $U%$H.MIL@relay.cs.net
.GOV                       $U%$H.GOV@relay.cs.net
.ORG                       $U%$H.ORG@relay.cs.net
.NET                       $U%$H.NET@relay.cs.net
!
! Esse ultimo conjunto de regras poderia ser substituido
! pela regra que casa com tudo ("default"):
! .                          $U%$H@relay.cs.net
! o preco pago por essa simplificacao 'e que denominacao
! com dominios ilegais nao seriam rejeitadas no sistema local.
! A mensagem seria passada para o sistema retransmissor e so
! entao rejeitada e retransmitida de volta com indicacao de
! erro.
!
! Atencao para a linha em branco a seguir, que separa as regras
! da tabela de canais.

!*** Tabela de Canais
! (Alem de separar as regras de reescrita da tabela de canais,
! linha em branco tambem separam os blocos de canais dentro
! da tabela de canais)
!
! Canal Local
1
hospedeiro.dominio

! Canal Decnet
d 733
DECNET-MAIL
! O atributo 733 usado com os canais DECNET forca o truncamento

```

```

! dos nomes estilo dominio apos o primeiro ponto. 'E util
! quando os nomes-DECNET dos n'os remotos correspondem 'a
! primeira parte do nome estilo dominio.
! Para o caso em que os nomes-DECNET nao correspondem 'a
! primeira parte do nome estilo dominio usa-se a facilidade
! das linhas adicionais: onde o "nome-hospedeiro" corresponde
! ao sistema que deve usar esse canal e o "nome-conveniente"
! o nome-DECNET do hospedeiro que sera usado no "To:" do
! envelope da mensagem.

```

```

CLARMATH.CLAREMONT.EDU      CMATH
MEDCHEM.CLAREMONT.EDU      MEDCHM

```

```

! Canal Phonenet
p periodic network
relay.cs.net

```

```

a_canal 822

```

```

a
e

```

```

b_canal 733 network

```

```

b
f

```

4.2 Informações sobre o Despachador Sendmail

Sendmail [Sun90a] implementa uma facilidade geral de transporte interrede. Sendmail não interfaceia com o usuário e nem realmente entrega a mensagem. Ao invés, coleta a mensagem de um programa tal como *mail* ou *elm*, edita a mensagem conforme requerido pelo despachador destino, e chama os despachadores apropriados para fazer entrega ou enfileiramento para transmissão pela rede. São usadas três tipos de interface tanto para submeter mensagem ao Sendmail quanto para este enviar mensagem para o despachador: o convencional "Argument Vector/Exit Status" do Unix, SMTP sobre "pipes" e SMTP sobre uma conexão TCP.

A configuração do Sendmail é controlada pela leitura de um arquivo de configuração quando este inicia a rodar. Esse arquivo de configuração possui: *definição de macros* (define valores de macros usadas internamente), *declaração de cabeçalhos* (o formato das linhas de cabeçalho que serão processadas especialmente e linhas que serão acrescentadas ou reformatadas), *definição de despachadores* (informação tais como a localização e característica de cada despachador), *regras de reescrita* (sistema de regras para reescrever nomes), e *opções* (tais como: "pathname" para vários arquivos de suporte, "timeouts", modos assumidos, etc.)

Como esse arquivo de configuração foi idealizado para ser lido por máquina, a leitura desse arquivo por um ser humano é bastante árdua. Existem softwares públicos, obtidos na base SIMTEL20, que ajudam nesse sentido tais como EASE, que permite escrever numa linguagem de mais alto nível e gerar um *sendmail.cf*, e CFC, para colocar um *sendmail.cf* numa forma mais legível que usa uma linguagem de mais alto nível. No entanto, após certo traquejo com o vocabulário usado no *sendmail.cf*, essas ferramentas

Tabela 4.1: Prefixos X Tipo de Canal

<i>prefixo</i>	<i>tipo de canal</i>
ANJE_	ANJE (BITNET)
BIT_	JNET (BITNET)
CN_	Rede australiana nacional interna
CTCP_	TCP/IP da Universidade Carnegie Mellon
DIR_	Canal diretório
DN_	Phonenet em cima de DECNET
DSMTP_	SMTP em cima de DECNET
ETCP_	TCP/IP da Excelan
FTCP_	TCP/IP da Network Research Co. FUSION
KER_	Canal que usa o protocolo Kermit
MAIL_	Canal geral de entrega de MAIL-VMS
MTCP_	TCP/IP da TGV Multinet
PX25_	Phonenet sobre X.25
TCP_	TCP/IP da Tektronix
UTCP_	TCP/IP de conexão ULTRIX
UUCP_	Canal UUCP
WTCP_	TCP/IP da Wollongong (WIN/TCP)
XSMTP_	SMTP sobre X.25

Tabela 4.2: Chaves modificadoras opcionais

<i>chave</i>	<i>função</i>
733	usa encaminhamento com % no envelope
733_header	usa encaminhamento com % no cabeçalho
822	usa encaminhamento fonte no envelope
822_header	usa encaminhamento fonte no cabeçalho
bangstyle	usa encaminhamento com ! (UUCP) no envelope
bidirecional	o canal é servido por ambos: mestre e escravo
daemon	o nome que segue a chave especifica o nome de um usuário associado ao "daemon" que manipula correio no sistema
exproute	encaminhamento explícito (coloca o nome do sistema local) para denominações "From:" desse canal
grey	usa formato do GREY BOOK (domínios em ordem reversa)
headerbotton	cabeçalho no final da mensagem (apenas VMS-MAIL)
headeromit	cabeçalho omitido da mensagem (apenas VMS-MAIL)
immediate	entrega iniciada imediatamente após colocação na fila
logging	registra a colocação e a retirada de mensagens da fila
master	o canal é servido apenas por um programa mestre
master_debug	habilita depuração no programa mestre
multigate	o canal serve múltiplos entroncamentos BITNET
multiple	aceita múltiplos hospedeiros numa única cópia de mensagem
mx	para redes TCP/IP que suportam repasse de correspondência
network	exige que o usuário tenha privilégio NETMBX (no VMS)
nolocaluser	não usa nomes de usuário locais para checagem de de autorização numa conexão remota
nomx	para redes TCP/IP que não suportam repasse de mensagem
nosendpost	não envia cópias de mensagem com falha ao postmaster
nsmtpt	o canal não usa SMTP
nox_env_to	não insere linhas X_Envelope_to:
percents	usa encaminhamento com % no envelope
period	o que segue a palavra especifica o período que o canal é servido
periodic	o canal é servido apenas periodicamente
queue	segue o nome da fila de "jobs" ("batch jobs") onde o mestre roda. A assumida é MAIL\$BATCH
sendpost	envia cópias de falha ao postmaster
single	uma cópia de mensagem para cada denominação
single_sys	uma cópia de mensagem para cada sistema
slave	o canal é servido por um programa escravo
slave_debug	habilita a depuração do programa escravo
smtpt	o canal usa SMTP
source route	usa encaminhamentos fonte no envelope
user	usa nomes de usuários locais para checagem de de autorização numa conexão remota
uucp	usa encaminhamento UUCP, com exclamação, no envelope
x_env_to	insere X_Envelope_to: enquanto enfileira

podem ser dispensadas. Como a construção do arquivo de configuração a partir do 0 é complexa, a própria Sun sugere ajustar um arquivo de configuração existente, por ela fornecido, às suas necessidades. Esse arquivo fornecido pela Sun é mostrado na pag.64, e os inúmeros comentários ajudam a identificar as estruturas de sintaxe descritas a seguir.

O arquivo de configuração é organizado como uma série de linhas, cada qual começando com um único caracter que define a semântica do resto da linha. Linhas começando com espaço ou <tab> são linhas de continuação. Linhas em branco ou linhas com o símbolo “#” são comentários.

D - define macro

Macros são nomeadas com um único caracter, que é selecionado do conjunto ASCII. As macros definidas pelo usuário são nomeadas com letras maiúsculas. As com letras minúsculas e símbolos especiais são usados internamente.

A sintaxe para definição de macro é:

`Dxval`

onde *x* é o nome da macro e *val* é o valor que terá. Essa macro pode ser inserida em vários lugares no arquivo através da sequência `$x`.

C e F - definem classes

Pode-se definir classes de palavras afim de casarem (fazer correspondência) com o lado esquerdo das regras de tradução. Por exemplo, pode-se criar classe de todos os nomes locais afim de eliminar a tentativa da máquina enviar a si mesma.

Essas classes podem ser definidas no próprio arquivo de configuração, lidas de outro arquivo ou de outro comando. Os nomes das classes devem ser do conjunto de letras maiúsculas, as minúsculas e caracteres especiais são reservadas para uso interno.

A sintaxe é:

`Cc palavra1 palavra2 ... palavraN`

`Fc arquivo`

`Fc -- comando`

A primeira forma define que a classe “c” pode ser qualquer das palavras 1 e 2 ...N. A segunda lê as palavras de um arquivo. A terceira executa um comando e lê os elementos da classe da saída padrão do comando, ex:

`FC -- awk 'print $2' /etc/hosts`

Pode-se dividir as linhas de classe em múltiplas linhas. Por exemplo, as duas formas:

`CH monet ucbmonet`

ou

`CH monet`

`CH ucbmonet`

são equivalentes.

O - estabelece opção

Existem várias opções (que não devem ser confundidas com “flags dos despachadores” ou argumentos de linha de comando) que podem ser estabelecidas no arquivo de configuração. Opções são representadas com um único caracter. A sintaxe é:

Oovalor

Que estabelece o “valor” para a opção “o”. Dependendo da opção, o valor pode ser uma cadeia de caracteres, um inteiro, um booleano (com valores legais t,T,f ou F - o assumido é TRUE), ou um intervalo de tempo.

O trecho “Opções” no arquivo à pag.64 mostra a lista de opções possíveis.

P - definição de precedência

Pode-se estabelecer valores para o campo “Precedence:” usando a linha de controle P. A sintaxe para esse campo é:

Pnome=num

onde o nome é encontrado no campo “Precedence:”, e a classe da mensagem é estabelecida para “num”. Altos números significam precedência mais alta. Números menores que zero têm a propriedade especial de não retornar mensagens de erro. A precedência assumida é zero. Por exemplo:

Pfirst-class=0

Pspecial-delivery=100

Pjunk=-100

T - usuários de confiança

Usuários aos quais é permitido sobrepor o nome do que envia usando a flag -f na linha de comando. Esses tipicamente são, “root”, “uucp” e “network”. A sintaxe é

T usuário1 usuário2 ...

Pode-se ter mais do que uma linha dessas.

H - define cabeçalho

Os formatos das linhas de cabeçalho são definidos pelas linhas “H”. A sintaxe é:

H[?mflags?]hnome:hmáscara

Linhas de continuação são inseridas diretamente na mensagem que sai. A “hmáscara” é macro que é expandida antes de ser inserida na mensagem. Se a expansão é vazia, a linha de cabeçalho não é incluída. Se “mflags”(limitadas por interrogações) são especificadas, pelo menos uma das flags deve estar estabelecida na definição do despachador para esse cabeçalho ser colocado automaticamente. Se um desses cabeçalhos está na entrada, esse é dirigido para saída desconsiderando essas flags.

R e S - regras de reescrita

A análise da denominação é feita de acordo com regras de reescrita, que são simplesmente um sistema de casamento de padrões. O sendmail percorre o conjunto de regras procurando casar o nome com o lado esquerdo da regra. Quando casa o nome é substituído pelo lado direito da regra.

Existem vários conjuntos de regras. Alguns conjuntos são usados internamente e possuem semânticas específicas. Outros conjuntos não possuem semântica determinada e podem ser referenciados pela definição de despachadores ou por outros conjuntos de regras.

A sintaxe é:

Sn

estabelece que será iniciado o conjunto de regras **n**. Se começa-se um conjunto mais do que uma vez, a definição anterior é apagada.

E a sintaxe de cada regra é:

Rlesq rldir comentários

O *lesq* é o padrão aplicado à entrada. Se casa a entrada é reescrita pelo *ldir*. Comentários são ignorados.

M - define despachador

Programas e interfaces com despachadores são definidos pela sintaxe:

Mnome,campo=valor*

onde “nome” é o nome do despachador (usado nas mensagens de erro) e o par “campo=valor” define atributos do despachador. Os campos são:

Path especificação do diretório para ativar o despachador

Flags flags especiais para o despachador

Sender conjunto de regras para os nomes remetentes

Recipient conjunto de regras para os nomes destinos

Argv vetor de argumentos para passar para o despachador

Eol cadeia de fim de linha para o despachador

Maxsize comprimento máximo da mensagem para o despachador

Length comprimento máximo do Argv para o despachador

Apenas o primeiro caracter do nome do campo é checado.

A seguir serão descritos os detalhes das regras de reescrita e descrição dos despachadores.

A Semântica

Algumas macros devem ser definidas para transmitir informação para o Sendmail. Essas são listadas na tabela 4.3.I. A macro “\$e” é “impressa” quando a sessão SMTP começa. A primeira palavra da “\$e” deve ser a macro “\$j”. A macro “\$j” deve estar no formato de domínios. A macro “\$o” consiste de uma lista de caracteres que serve como e para separar tokens. Por exemplo, se “y” está na macro “\$o”, então a entrada “xyzyz” será interpretada como quatro tokens: “x”, “y”, “zz” e “y”. Finalmente a “\$q” especifica como um nome remetente deve aparecer na mensagem. Reportando-se ao arquivo da pag.64 a macro “\$y” é definida Dq\$q\$?x (\$x)\$. Uma alternativa seria Dq\$?x\$x \$.<\$g>. Esses formatos correspondem a:

nowicki@sun.com (Bill Nowicki)

e

Bill Nowicki <nowicki@sun.com>

Algumas macros são definidas pelo Sendmail para uso nos argumentos dos despachadores ou para outros contextos. Essas são listadas na tabela 4.3.II.

Existem três tipos de datas que podem ser usadas: “\$a” é a extraída da linha “Date:” da mensagem, “\$b” é a corrente usada para registro. Se não há a linha “Date:” na mensagem, “\$a” também será a corrente. A macro “\$d” equivale a “\$a” no formato UNIX (ctime).

A macro “\$f” é o identificador do remetente original. Quando se envia para um hospedeiro determinado a macro “\$g” é o nome do remetente relativo ao recipiente. Por exemplo, se “eric” envia para “bollard@matisse” da máquina “ucbarpa”, a macro “\$f” será “eric” e a “\$g” será “eric@ucbarpa”.

A macro “\$x” é o nome completo do remetente, que pode ser determinado de várias maneiras. Pode ser passado como flag ao Sendmail, como valor da linha de cabeçalho “Full-Name:” se existir, e como comentário na linha “From:”. Se todas essas falham e a mensagem foi originada localmente, o nome completo é buscado no arquivo /etc/passwd.

As macros “\$p” e “\$t” são usadas para criar cadeias únicas (por ex., para o campo “Message-Id”). A macro “\$i” é o identificador da fila no hospedeiro local, se colocada na linha de carimbo de tempo será útil para seguir a trilha da mensagem. A macro “\$v” é o número da versão do Sendmail, é colocado normalmente nos carimbos de tempo e é extremamente útil para depuração. A macro “\$w” é o nome primário do hospedeiro local. A macro “\$c” especifica o número de vezes que a mensagem foi processada. Pode ser determinado pelo número de carimbos de tempo na mensagem. As “\$r” e “\$s” são estabelecidas com o protocolo usado para comunicar com o Sendmail e com o nome do hospedeiro que envia.

Pode-se especificar condicionais usando a sintaxe: “\$?xtexto1\$—texto2\$”. Que insere o texto1 se a macro “\$x” é estabelecida e texto2 caso contrário. A cláusula senão (\$—) pode ser omitida.

A classe “\$=w” é o conjunto de todos os nomes que o hospedeiro local é conhecido. Pode ser usado para apagar nomes locais. A classe “\$=m” é estabelecida para o nome do domínio.

Tabela 4.3: Macros Minúsculas

I. Macros para transmitir informação ao Sendmail   e mensagem de inscrição SMTPb domínio oficial para este localc formato da linha "From" do UNIXn nome do daemon (p/ mensagens de erro)o o conjunto de "separadores" nos nomesq formato assumido para os nomes remetentes II. Macros usadas nos argumentos dos despachadores ou outros contextos   a data original no formato ARPANETb data corrente no formato ARPANETc contador de saltos (hop count)d data no formato UNIX (ctime)f o nome do remetente (from)g o nome do remetente relativo ao destinatárioh o hospedeiro destinatárioi o ID da filam nome do domíniop ID do processo Sendmailr protocolo usados nome do hospedeiro remetentet representação numérica da data correnteu usuário recipientev o número da versão do Sendmailw o nome do hospedeiro localx o nome completo do remetentez o nome do diretório "home" do recipiente

Lado Esquerdo

O lado esquerdo das regras de reescrita contém um padrão. Palavras normais são casadas diretamente. O sinal “\$” introduz meta-símbolos, que casam coisas diferentes de meras palavras, tais como classes e macros. Os meta-símbolos estão listados na tabela 4.4.I. Se casa algum padrão, este é atribuído ao símbolo “\$n” para substituição no lado direito da regra. Por exemplo, se o lado esquerdo “\$-:\$+” é aplicado à entrada “UCBARPA:eric”, os valores passados para o lado direito da regra será “UCBARPA” para “\$1” e “eric” para “\$2”.

A “\$%x” usa a macro “x” para especificar o nome de um mapa do NIS. A forma especial “\$%y” corresponde a qualquer nome de hospedeiro no mapa “hosts.byname”, ou no /etc/hosts se o NIS não estiver rodando.

Lado Direito

Quando casa o lado esquerdo de uma regra de reescrita, a entrada é substituída pelo lado direito da regra. Os “tokens” são copiados diretamente a menos que comecem com “\$”. Os meta-símbolos para complicadas substituições no lado direito da regra são listados na tabela 4.4.II.

A sintaxe “\$n” (n sendo um dígito) substitui o correspondente valor “\$+”, “\$-”, “\$*”, “\$=” casado no lado esquerdo. Pode ser usado em qualquer lugar.

A sintaxe “\$>n” faz com que o resto da linha seja substituído normalmente e então passado para o conjunto de regras n. O valor final do conjunto de regras n tornar-se-á a substituição dessa regra. (Pode ser comparado com uma chamada de função ou procedimento).

A sintaxe “\$#” deve ser usada apenas no conjunto de regras “0”. Faz com que a avaliação do conjunto de regras termine imediatamente, e sinaliza ao Sendmail que o nome foi completamente determinado. A sintaxe completa para a especificação a tripla {despachador, hospedeiro, usuário}, que é necessária para encaminhar ao despachador é: “\$despachador\$@hospedeiro\$:usuário”. Pode haver mais processamento dependendo do despachador, por exemplo, nome local é substituído por um apelido.

Afim de controlar a avaliação, o lado direito pode também ser precedido por “\$@” e “\$:”. O prefixo “\$@” faz com que o resto do lado direito retorne como valor do conjunto de regras. O prefixo “\$:” faz com que a regra termine imediatamente, mas o conjunto de regras continue, podendo ser usado assim para limitar a regra a uma aplicação. Os prefixos “\$@” e “\$:” podem preceder a especificação “\$>”. Por exemplo: R\$+ \$:\$>7\$1, casa qualquer coisa, passa para o conjunto de regras 7 e continua. O “\$:” é necessário para evitar um loop infinito. A sintaxe \$[hospedeiro]\$ substitui o nome do hospedeiro com o nome oficial ou primário que é o primeiro no mapa “hosts.byname” do NIS ou no “/etc/hosts” se o NIS não estiver rodando. É usado para eliminar apelidos para hospedeiros. A sintaxe \$x nome\$ substitui a cadeia pelo resultado do mapa do NIS indicado na macro x.

Semântica dos Conjuntos de Regras

Existem cinco conjuntos de regras com semânticas específicas. A relação entre eles é mostrada na fig. 4.1.

Tabela 4.4: Meta-Símbolos

I. Meta-símbolos usados no lado esquerdo das regras de reescrita**\$*** casa zero ou mais tokens**\$+** casa um ou mais tokens**\$-** casa exatamente um token**\$=x** casa qualquer cadeia na classe x**\$ x** casa qualquer token que não esteja na classe x**\$_x** casa qualquer token no mapa \$x do NIS**\$x** casa macro x**II. Meta-símbolos usados no lado direito das regras de reescrita****\$x** expande a macro x**\$n** substitue o token indefinido n do lado esquerdo da regra**\$>n** chama o conjunto de regras n**\$#despachador** determina para o despachador**\$@hospedeiro** especifica hospedeiro (+ prefixo - retorno de conjunto de regra)**\$:usuário** especifica usuário(+ prefixo - limita regra)**\$hospedeiro\$** mapeia para o nome de hospedeiro primário**\$rnome\$** mapeia nome através do mapa \$r do NIS

Figura 4.1: Relação entre os conjuntos de Regras

O conjunto de regras “3” coloca o nome numa forma canônica, cuja básica é: parte-local@especificação-hospedeiro-domínio. Se não é especificado o sinal “@”, então a “especificação- hospedeiro-domínio” pode ser acrescentada com a do remetente (se a flag C está estabelecida na definição do despachador remetente). O conjunto “3” é aplicado pelo Sendmail antes de qualquer coisa ser feita com o nome.

O conjunto de regras “0” é aplicada após o “3”, a nomes que realmente vão especificar recipientes. Deve determinar a tripla despachador, hospedeiro, usuário. O despachador deve estar definido nas definições para despachador no arquivo de configuração. O hospedeiro é definido na macro “\$h” usada para expansão de argumento no despachador específico, o usuário é definido na macro “\$u”.

Após a aplicação do conjunto de regras “1” e “2” , respectivamente, a todos os nomes “From:” e “To:/Cc:”, são aplicadas, também respectivamente, as regras “S=” e “R=” especificadas na definição do despachador. Note-se que isso é feito várias vezes para uma mensagem, dependendo de quantos despachadores a mensagem foi encaminhada pela regra “0”.

O conjunto de regras “4” é aplicado por último a todos os nomes na mensagem. É tipicamente usado para traduzir da forma interna para externa.

O despachador como o nome especial “error” pode ser usado para gerar um erro ao usuário. O campo usuário é a mensagem que será mostrada. Por exemplo:

```
$#error$: Host unknown in this domain
```

no lado direito de uma regra faz com que seja gerado esse erro se o lado esquerdo casa com a entrada. Esse despachador tem função apenas no conjunto de regras “0”.

Semântica da Descrição dos Despachadores

Cada despachador tem um nome interno, que pode ser arbitrário, exceto os nomes “local” e “prog” que devem ser definidos, respectivamente em primeiro e segundo lugares.

O “pathname” do despachador deve ser dado no campo P. Se o despachador é acessado via uma conexão TCP, usa-se ao invés, a cadeia “[TCP]”.

O campo F define as flags do despachador, listadas na tabela 63.

Os campos S e R, descrevem, respectivamente, o número do conjunto de regras a ser aplicado nos nomes remetentes e recipientes (destinatários).

O campo A dá a máscara do vetor de argumentos do despachador.

Por exemplo, a especificação:

```
Mlocal, P = /bin/mail, F = rlsmi, S = 10, R = 20, A = mail -d $u
Mether, P = [TCP],      F = mec,   S = 11, R = 21, A = TCP $h, M = 100000
```

especifica um despachador para entrega local e um despachador para entrega ethernet.

O primeiro chamado “local”, é localizado no arquivo /bin/mail, aspas devem ser retiradas dos nomes e entregas podem ser feitas a múltiplos usuários de uma vez. O conjunto de regras “10” deve ser aplicado aos nomes remetentes na mensagem e o conjunto de regras “20” aos nomes recipientes (destinatários). O vetor de argumento para enviar a mensagem será a palavra “mail”, a palavra “-d”, e as palavras contendo os nomes dos usuários recipientes (destinatários).

O segundo despachador chamado “ether”, será conectado via TCP, manipula muitos usuários por vez, conexões deverão ser adiadas, e o domínio do nome do remetente deve ser acrescentado a qualquer nome recipiente (destinatário) sem domínio. Os nomes remetentes serão processados pelo conjunto de regras “11” e os nomes recipientes pelo conjunto de regras “21”. Existe um limite de 100.000 bytes para mensagens processadas através desse despachador.

Flags para descrição dos despachadores

C se a mensagem é recebida de um despachador com essa flag estabelecida, quaisquer nomes no cabeçalho que não possuem o sinal “@” após ser reescrito pelo conjunto de regra “3”, terá a cláusula “@domínio” do remetente, a ele acrescentada. Isso permite que cabeçalhos na forma

```
From: usuárioa@hospedeiroa
To: usuáriob@hospedeirob, usuárioc
serem reescritos como:
From: usuárioa@hospedeiroa
To: usuáriob@hospedeirob, usuárioc@hospedeiroa
automaticamente
```

D esse despachador quer a linha de cabeçalho “Date:”

e esse despachador é caro para se conectar, assim tenta-se evitar a conexão normal; qualquer conexão ocorrerá durante um processamento de fila.

F o despachador quer a linha de cabeçalho "From:"

h letras maiúsculas deverão ser preservadas nos nomes de hospedeiro desse despachador

L limita o comprimento das linhas como especificado no RFC'821

l esse despachador é local (ou seja, a entrega final será efetuada)

M esse despachador quer a linha de cabeçalho "Message-Id:"

m o despachador pode enviar para muitos usuários num mesmo hospedeiro numa transação. Quando a macro "\$u" ocorre na parte *argv* do despachador, esse campo será repetido o quanto necessário para todos os usuários qualificados. O campo "L =" da descrição do despachador pode ser usado para limitar o comprimento total da expansão da macro "\$u".

n não inserir a linha "From:" estilo-Unix em frente da mensagem

P esse despachador quer a linha "Return-Path:"

p sempre acrescenta o nome de hospedeiro na linha "MAIL From:" do SMTP, mesmo se já existir algum.

S não estabelece o "userid" antes de chamar o despachador. Poderia ser usado num ambiente de segurança onde o Sendmail roda como "root". Poderia ser usado para evitar nomes forjados

s retira aspas do nome antes de chamar o despachador

U Esse despachador quer linhas "From:" estilo-Unix, com o horrível estilo-UUCP "remote from <host>" no final.

u letras minúsculas deverão ser preservadas nos nomes de usuários nesse despachador.

X usa o algoritmo do ponto escondido como especificado no RFC'821; basicamente toda linha começando com ponto terá um ponto extra. Isto assegura que linhas na mensagem contendo ponto não terminarão a mensagem prematuramente.

x esse despachador quer a linha de cabeçalho "Full-Name:"

```
#####
#
# Arquivo de configuracao do Sendmail para
# "MAQUINAS PRINCIPAIS"
# Deve-se instalar este arquivo como /etc/sendmail.cf
# se a sua maquina 'e o principal (ou somente)
# retransmissor de correspondencia do seu dominio.
# Entao edite o arquivo para adaptar a configuracao
# de sua rede.
#
```

```

# @(#)main.mc 1.17 90/01/04 SMI
#

### local info

# meu nome de hospedeiro oficial
# Voce tem duas opcoes aqui. Se voce quer que a maquina "entron-
# camento" identifique-se como dominio use:
#Dj$m
# Se voce quer que a maquina "entroncamento" apareca dentro do dominio, use:
Dj$w.$m
# Se possuir um nome de hospedeiro completamente qualificado, use:
#Dj$w

# principal despachador-retransmissor (mailer-relay) - a escolha tipica
# 'e "ddn" se voce esta na Defense Data Network (e.g. Arpanet or Milnet)
DMsmartuucp

# principal hospedeiro retransmissor: usa o $M despachador para enviar
# correspondencia para outros dominios
DRddn-gateway
CRddn-gateway

# Se voce quer carregar os hospedeiros que permitem correspondencia, use:
# FS /usr/lib/mailhosts
# e entao troque as ocorrencias de $%y para $=S.
# Senao, o mapeamento hosts.byname do NIS \'{e} assumido se o NIS
# estiver rodando (senao estiver, o arquivo/etc/hosts).

# dominios de nivel topo validos
CT arpa com edu gov mil net org
CT us de fr jp kr nz il uk no au fi nl se ca ch my dk ar

# opcoes provavelmente desejadas no hospedeiro de correspondencia

# ponto de checagem da fila apos esse nro de recipientes(destinatarios)
OC10

# recusa o envio de mensagens vazias para mais do que esse nro de recipientes
Ob10

#####
#
# Informacao de Configuracao Geral

# nomes de dominio local
#
# Se o dominio do NIS 'e diferente do nome que voce deseja que apareca
# nos cabecinhos de correspondencia, edite as linhas abaixo para refle-
# tir os dominios que voce deseja.
# Note que o primeiro componente do nome de dominio do NIS 'e retirado
# a menos que o nome de dominio comece com um "." ou "+".
# Exemplo:

```

```
# DmCS.Podunk.EDU
# Cm cs cs.Podunk.EDU
#
# O Dm 'e o valor estabelecido para as correspondencias que saem.
# o Cm 'e o valor que 'e aceito para as correspondencias que chegam.
# Normalmente sao os mesmos, porem voce pode querer possuir mais de
# uma linha Cm para reconhecer mais do que um nome de dominio durante
# uma transicao.
```

```
# Numero da versao do arquivo de configuracao
DVSMI-4.1
```

Macros Padroes

```
# nome usado para enviar mensagens com erro
DnMailer-Daemon
# formato de cabeçalho do UNIX
DlFrom $g $d
# caracteres delimitadores (operadores)
Do.:%@!~/[]
# formato de um nome completo
Dq$g$?x ($x)$
# Mensagem usada para o "login" do SMTP
De$j Sendmail $v/$V ready at $b
```

Opcoes

```
# Modo Remoto - envia pelo servidor se o diretorio caixa-postal esta' montado
OR
# localizacao do arquivo de apelidos
OA/etc/aliases
# modo de entrega assumido (entrega em "background")
Odbackground
# reconstruo o arquivo de apelidos automaticamente
OD
# modo de protecao dos arquivos temporarios -- 0600 para seguranca, 0644 para permissao
OF0600
# GID (identificacao de grupo) assumido
Og1
# localizacao do arquivo de ajuda
OH/usr/lib/sendmail.hf
# nivel de registro
OL15
# mensagens assumidas - estilo antigo
Do
# "Postmaster" local - gerente de correspondencia local (p/ replicas de erro)
OPPostmaster
# diretorio da fila de correspondencia
OQ/usr/spool/mqueue
# tempo de expiracao para leitura no protocolo SMTP
Or15m
# arquivo de estado - nenhum
```

```

OS/etc/sendmail.st
# por seguranca enfileira tudo antes de iniciar a transmissao
Os
# apos ficarem enfileiradas esse tempo, as mensagens retornam ao
# que enviou
OT3d
# UID (identificacao de usuario) assumido
Ou1

### Precedencia de Mensagens
Pfirst-class=0
Pspecial-delivery=100
Pjunk=-100

### Usuarios de Confianca
T root daemon uucp

### Formato dos cabecalhos
H?P?Return-Path: <$g>
HReceived: $?sfrom $s $.by $j ($v/$V)
id $i; $b
H?D?Resent-Date: $a
H?D?Date: $a
H?F?Resent-From: $q
H?F?From: $q
H?x?Full-Name: $x
HSubject:
H?M?Resent-Message-Id: <$t.$i@$j>
H?M?Message-Id: <$t.$i@$j>
HErrors-To:

#####
### Regras de Traducao ###
#####

# Pre-escrita do campo Remetente
S1
# Nada e' necessario.

# Pre-escrita do campo Destinatario
S2
# Nada e' necessario.

# Colocacao dos nomes na forma Canonica

# O formato interno dos nome dentro das regras de reescrita e':
#   qualquer_coisa<@hospedeiro.dominio.dominio...>qualquer_coisa
# Coloca-se todo tipo de nome nesse formato, exceto para nomes
# locais, que nao possuem parte de hospedeiro. A razao para "<>"
# e' que todo nome de hospedeiro relevante pode estar em frente
# do nome (para encaminhamento fonte), ou atras (forma normal).
# Coloca-se o que e' para encaminhar dentro de "<>" para ser

```

facil de achar.

#

S3

tratando o caso especial "from:<>"

R\$*<>\$* \$00 transforma no "token" magico

canonicalizacao de texto basica

R\$*<\$+>\$* \$2 analise RFC 822 basica

para certificar que a sintaxe <@a,@b,@c:user@d>

e' facil de analisar - desfaz depois

R0\$+, \$+:\$+ @1:\$2:\$3 troca toda "," por ":"

R0\$+:\$+ @0\$>6<@1>:\$2 encaminhamento fonte canonico

R\$+:\$+;@+ \$0\$1:\$2;@3 sintaxe de lista

R\$+@+ \$:\$1<@2> enfase no dominio

R\$+<\$+@+> \$1\$2<@3> move o enfoque

R\$+<@+> @0\$>6\$1<@2> ja canonico

conversao de nomes estilo antigo para nomes estilo dominio

Todos os nomes estilo antigo sao analisados da esquerda para

direita, sem precedencia.

R\$-!\$+ @0\$>6\$2<@1.uucp> hospedeiro_uucp!usuario

R\$-.\$+!\$+ @0\$>6\$3<@1.\$2> hospedeiro.dominio!usuario

R\$+/%\$+ @0\$>3\$1@2 usuario%/hospedeiro

Reescrita final para sair

S4

R\$+<@+ .uucp> \$2!\$1 u@h.uucp => h!u

R\$+ \$: \$>9 \$1 limpa denominacao

R\$*<\$+>\$* \$1\$2\$3 desfaz "<>"

Limpa o nome para passar para algum despachador

(mas deixa com "<>"

S9

R\$=w!@ \$0\$w!\$n

R0 \$0\$ n trata do erro na denominacao: <>

R\$*<\$*LOCAL>\$* \$1<\$2\$m>\$3 troca informacao local

R<@+>+*:\$+:\$+ <@1>\$2,\$3:\$4 <encaminhamento> canonico

#####

Regras de Reescrita

conversoes especiais locais

S6

#R\$*<@*\$=m>\$* \$1<@2\$LOCAL>\$4 converte para o dominio LOCAL

Especificacao dos despachadores local e de programa


```
Mlocal, P=/bin/mail, F=rlsDFMmnP, S=10, R=20, A=mail -d $u
Mprog, P=/bin/sh, F=lsDFMeuP, S=10, R=20, A=sh -c $u
```

```
S10
```

```
# Nada e' necessario
```

```
S20
```

```
# Nada e' necessario
```

```
#####
```

```
####
```

```
#### Especificacao do Despachador Ethernet
```

```
####
```

```
#### Mensagens processadas por essa configuracao estao assumidas
```

```
#### permanecerem no mesmo dominio. Realmente nao ha nada a fazer
```

```
#### particularmente com Ethernet - o nome e' historico
```

```
Mether, P=[TCP], F=msDFMuCX, S=11, R=21, A=TCP $h
```

```
S11
```

```
R$*<@ $+>$* $Q$1<@ $2>$3 ja correto
```

```
R$+ $Q$1<@ $w> acrescenta nosso nome de hospedeiro
```

```
S21
```

```
# Nada e' necessario
```

```
#####
```

```
# Codigo geral para converter de volta para nomes UUCP,
```

```
# estilo antigo
```

```
S5
```

```
R$*<@LOCAL> $Q $w!$1 nome@LOCAL => sun!nome
```

```
R$*<@$.LOCAL> $Q $2!$1 u@h.LOCAL => h!u
```

```
R$*<@$.uucp> $Q $2!$1 u@h.uucp => h!u
```

```
R$*<@ $+> $Q $2!$1 u@h => h!u
```

```
# encaminhamentos nao tem sentido aqui
```

```
R<@ $+>$* $Q @ $1$2 desfaz <>
```

```
R$*<$+>$* $Q $1$2$3 desfaz <> estranho
```

```
# Especificacao do Despachador UUCP
```

```
Muucp, P=/usr/bin/uux, F=msDFMhuU, S=13, R=23,
```

```
A=uux - -r -a $f $h!rmail ($u)
```

```
# Conversao do campo remetente UUCP (From)
```

```
S13
```

```
R$+ $: $>5$1 converte para o estilo antigo
```

```
R$=w!$+ $2 tira o nome local
```

```
R$+ $: $w!$1 insere o nome de hospedeiro real
```

```
# Conversao dos campos destinatarios UUCP (To,Cc)
```

```
S23
```

```
R$+ $: $>5$1 converte para o estilo antigo
```

```
#####
#
# Especificacao do Despachador DDN
#
# Envia a correspondencia para Defense Data Network
#   (tais como Arpanet ou Milnet)

Mddn, P=[TCP], F=msDFMuCX, S=22, R=22, A=TCP $h, E=\r\n

# mapa contendo o inverso do mail.aliases
DZmail.byaddr

S22
R$*<@LOCAL>$* $:$1
R$-<@> $:$3{Z$1@2$}   inverte apelidos
R$*<@>.$*>$* $@1<@2.$3>$4   ja correto
R$+<@>+>$* $@1<@2.$m>$3   acrescenta o nosso dominio
R$+ $@1<@m>   acrescenta o nosso dominio

# Despachador UUCP "esperto": usa o transporte UUCP, mas a denominacao
# estilo dominio
Msmartuucp, P=/usr/bin/uux, F=CmsDFMhuU, S=22, R=22,
A=uux - -r $h!rmail ($u)

#####
#
# Conjunto 0 de regras
#
# Este e' o conjunto de regras que determina o despachador
# para o qual a denominacao deve ser passada

# O conjunto de regras 30 chama o conjunto de regras 3 e entao o 0.
S30
R$* $: $>3 $1 primeira canonicalizacao
R$* $@ $>0 $1 e entao roda o conjunto de regras 0

S0
# Na entrada o nome ja foi posto na forma canonica pelo conjunto de regras 3
# tratando casos especiais
R@ $#local $:$n tratando da forma <>

# coloca o nome de hospedeiro local para LOCAL
R$*<$*=w.LOCAL>$* $1<$2LOCAL>$4 este_hospedeiro.LOCAL
R$*<$*=w.uucp>$* $1<$2LOCAL>$4 este_hospedeiro.uucp
R$*<$*=w>$* $1<$2LOCAL>$4 este_hospedeiro

# Correspondencia destinada ao entroncamento de dominio (US)
R$*<@LOCAL> $@$>30$1 retira noss nome e tenta de novo
R<@LOCAL>:$+ $@$>30$1 tenta de novo apos retirar encaminhamento
```

```

# Para especificacao numerica
R<@[$+]>:$* $:$>9 <@[$1]>:$2 limpa e entao continua...
R<@[$+]>:$* $#ether $@[$1] $:$2 especificacao numerica internet
R<@[$+]>,$* $#ether $@[$1] $:$2      especificacao numerica internet
R$*<@[$+]> $#ether $@[$2] $:$1      especificacao numerica internet

# entrega a hospedeiros ethernet conhecidos explicitamente especificados no
# nosso dominio
R$*<@[$y].LOCAL>$* $#ether $@[$2] $:$1<@[$2]>$3 usuario@hospedeiro.sun.com

# hospedeiro_ether.uucp e' tratado como hospedeiro_ether.$m aqui.
# Isso permite denominar de um local uucp foo!sun!hospedeiro_ether!usuario
R$*<@[$y].uucp>$* $#ether $@[$2] $:$1<@[$2]>$3 usuario@hospedeiro_ether.uucp

# Explicitamente especifica nomes no nosso dominio --
# os quais os nomes de hospedeiros nao sao conhecidos
R$*<@[$*.LOCAL]>$* $#error $:Never heard of host $2 in domain $m

# Limpa as denominacoes para uso externo -- retira LOCAL, encaminhamento ,=>:
R$* $:$>9 $1 entao continua...

# determina para denominacoes estilo UUCP
R<@[$-.uucp]>:$+ $#uucp $@[$1] $:$2 @hospedeiro.uucp:...
R$*<@[$-.uucp]> $#uucp $@[$2] $:$1 usuario@hospedeiro.uucp

# Passa apenas nomes com dominios conhecidos para o hospedeiro retransmissor
R$*<@[$*.$=T>$* $#M $@R $:$1<@[$2.$3]>$4 usuario@dominio.conhecido

# Passa os nomes validos ao hospedeiro retransmissor
R$*<@[$*.$+>$* $#M $@R $:$1<@[$2.$3]>$4 usuario@qualquer.dominio

# Se se esta' no DDN, comenta-se as duas linhas acima e usa, ao inves,
# o seguinte:
R$*<@[$*.$+>$* $#ddn $@ $2.$3 $:$1<@[$2.$3]>$4 usuario@qualquer.dominio

# Todas as denominacoes nas regras acima sao absolutas
# (qualificada completamente com dominios)
# As denominacoes abaixo podem ser parcialmente qualificadas
# entrega a hospedeiros ethernet conhecidos
R$*<@[$y]>$* $#ether $@[$2] $:$1<@[$2]>$3 usuario@hospedeiro_ether

# outros nomes nao locais que nao tem para onde ir; voltam ao remetente
R$*<@[$*.$->$* $#error $:Unknown domain $3
R$*<@[$+>$* $#error $:Never heard of $2 in domain $m
R$*@[$* $#error $:I don't understand $1@[$2

# Denominacoes locais com % nao sao realmente locais
R$+[%$+ $@>30$1@[$2 troca "%" por "@", e tenta de novo

# tudo o que sobra e' nome local
R$+ $#local $:$1 nomes locais

```

Considerações Finais

O conceito de *canal* no PMDF equivale ao de *despachador* ou *mailer* do Sendmail.

As regras de tradução no PMDF servem tanto para o "from:" quanto para o "to:". O canal é determinado quando após a aplicação de várias regras encontra-se uma em que o "sistema encaminhador" encontrado consta na tabela canal/hospedeiro no bloco de algum canal. Uma forma mais imediata e a fim de documentação é colocar, ao invés de regras de tradução, o nome do local destino direto na tabela canal/hospedeiro no bloco do canal que deve ser usado. Dependendo do canal utilizado a máquina que roda o PMDF pode sair com endereços de "from:" diferentes.

No Sendmail existem regras específicas para "from:", "to:", para determinar o despachador e, após este determinado, regras específicas do despachador. A sintaxe de cada conjunto de regras está descrito no arquivo de configuração, no entanto a aplicação dos vários conjuntos de regras depende de um encadeamento implícito como mostra a fig. 4.1. Assim a inclusão de uma regra no arquivo de configuração deve ser bem comentada para fins de manutenção, pois seu efeito não é bem claro apenas olhando o arquivo. No sistema implementado as configurações dos despachadores PMDF, mencionado em 6.5 e no apêndice C, e Sendmail, mencionado em 6.6 e nos apêndices D, E e F, foram projetados a fim de encaminhar as mensagens para o meio de transporte adequado conforme o endereço.

Capítulo 5

Protocolagem

Considerações Iniciais

Como foi visto no capítulo 3 entre as etapas envolvidas na estruturação de um sistema de correio eletrônico está a protocolagem. Como o sistema implementado envolve um ambiente heterogêneo, sentiu-se a necessidade de analisar os diferentes conjuntos de protocolos envolvidos, entre os quais se destacam: TCP/IP, que é enfocado em 5.1, DECNET abordado na seção 5.2 e UUCP que é objeto da seção 5.3. Além de cumprir a finalidade para o sistema implementado dá diretrizes para os aspectos a serem considerados numa rede heterogênea, podendo ser útil em implementações de outras aplicações tais como login remoto e transferência de arquivo.

5.1 Características do padrão TCP/IP

Para uma rede funcionar de forma correta, a informação originada num nó remetente deve ser transmitida por uma linha de comunicação e entregue ao receptor de forma inteligível. Devido à necessidade de interação de diversos tipos de hardware e software para se efetuar essa função, foi desenvolvido o conceito de família de protocolos de comunicação. Um exemplo é a família de protocolos Internet (*Internet Protocol Suite*). Ela está centrada em torno do *Internet Protocol* (IP), protocolo inter-rede. Outros membros dessa família são:

- *Transmission Control Protocol* (TCP), protocolo de controle de transmissão
- *User Datagram Protocol* (UDP), protocolo de datagrama de usuário
- *Address Resolution Protocol* (ARP), protocolo de determinação de endereço (endereço Internet — endereço Ethernet)
- *Reverse Address Resolution Protocol* (RARP), protocolo de determinação reversa de endereço (endereço Ethernet — endereço Internet)
- *Internet Control Message Protocol* (ICMP) protocolo de controle de mensagem inter-rede

A família como um todo é popularmente conhecida como TCP/IP, refletindo os nomes dos dois protocolos principais.

Tabela 5.1: Camadas do TCP/IP

Camada	Serviços de rede
Aplicação	Telnet, FTP, SMTP, DNS
Transporte	TCP, UDP, ICMP
Rede	IP
Enlace de dados	ARP, RARP, controlador (tal como Ethernet)
Física	cabo ou outro dispositivo.

A estrutura do TCP/IP pode ser vista como formada por uma série de camadas, como indicado na tabela 5.1

Entre os protocolos TCP/IP mais usados a nível de camada de aplicação estão:

Telnet : protocolo de acesso por terminal remoto

FTP : protocolo de transferência de arquivos

SMTP : protocolo de transferência de correspondência simples

DNS : serviço de mapeamento nome–endereço.

O protocolo DNS permite um mapeamento nome–endereço dinâmico e automático, de tal forma que a informação nome–endereço existe distribuída em bases nos vários domínios e subdomínios, e essas bases sejam pesquisadas usando esse protocolo. No sistema implementado não se está fazendo uso desse serviço, mas sim usando-se tabelas estáticas que contêm o mapeamento completo das máquinas envolvidas.

O TCP/IP encaminha um pacote pela rede de acordo com o endereço IP que foi a ele acrescentado pelo protocolo IP no hospedeiro que o originou. Para a operação correta do TCP/IP, uma rede deve possuir um endereço de rede IP, e todos os hospedeiros pertencentes a ela devem também possuir um endereço IP. O endereço IP de uma máquina consiste de:

- *Número de rede*, atribuído pelo NIC, e
- *Número de hospedeiro*, atribuído localmente.

Essa informação é representada num endereço de 32 bits, dividido em 4 campos de 8 bits. Cada campo é chamado de um octeto, e a informação neste é representada por um número decimal separado do próximo octeto por um ponto. O número decimal representa um byte do endereço Internet, e assume portanto valores de 0 a 255. Um exemplo típico de endereço é: 129.144.50.56

Os números de rede estão divididos em 3 classes: A, B e C, cada qual especificada pelo valor do primeiro octeto dentro de certo intervalo, como na tabela 5.2

Os endereços IP de classe A permitem poucas redes com grande número de hospedeiros. O NIC atribui endereços de classe A a grandes organizações ou universidades. Os de classe B provêm uma distribuição igual de número de redes e hospedeiros. O NIC atribui endereços de classe B a companhias de médio porte ou a instituições com vários hospedeiros nas várias redes. Os de classe C provêm muitas redes com poucos hospedeiros (no máximo 254).

Tabela 5.2: Classes dos números de rede Internet

Classe	Intervalo (prim. octeto)	Endereço da rede	Endereço do hospedeiro
A	0-127	xxx	xxx.xxx.xxx
B	128-191	xxx.xxx	xxx.xxx
C	192-223	xxx.xxx.xxx	xxx

Figura 5.1: Utilização de máscaras

Assim, conforme a classe conseguida junto ao NIC, uma instituição vê-se restrita a um certo número de hospedeiros.

Por exemplo, para endereços de classe B, é possível ter para hospedeiro números de 0.1 a 255.254 (os números 0 e 255 não são permitidos). Para melhor organização interna da instituição, seria interessante dividir esses números de hospedeiros entre as várias redes internas. Essa divisão é feita aplicando uma máscara nos octetos destinados a endereços de hospedeiros. Como essa divisão não é do conhecimento do NIC, é chamada de sub-rede. O encaminhamento dentro de uma rede com sub-redes é baseado no par número de rede/número de sub-rede. Fora da rede é baseado apenas no número de rede. A máscara de sub-rede é usada para dissecar um endereço IP nas partes número de rede, número de sub-rede e número de hospedeiro. A fig. 5.1 ilustra o efeito da máscara.

As máscaras são expressas como quatro octetos de números decimais (255.255.255.255) ou como um único número hexadecimal (0xFFFFFFFF). As máscaras assumidas para cada classe são:

Classe A : 255.0.0.0 (0xFF000000)

Classe B : 255.255.0.0 (0xFFFF0000)

Classe C : 255.255.255.0 (0xFFFFFFFF00)

O caso mais comum é aplicar uma máscara classe C numa rede classe B. Isto provê 256 sub-redes possíveis, cada uma podendo acomodar até 254 hospedeiros.

Num ambiente UNIX BSD, a base de dados que contém a máscara de rede assumida para o sistema é representada pelo arquivo `/etc/netmasks`. O arquivo é da forma:

```
#
# Base de dados com mascara de rede
# Apenas mascaras nao assumidas precisam ser definidas aqui
#
# Rede          Mascara
128.32.0.0      255.255.255.0
```

Para cada rede que possui sub-redes, existe uma linha nesse arquivo contendo o número de rede e a máscara a ser aplicada.

Máquinas encaminhadoras (de interface com mais de uma sub-rede) que possuem clientes NIS necessitam ter o arquivo `/etc/netmasks` além do mapa `netmasks.byaddr` do NIS. Isto é, durante a inicialização das máquinas na rede (*booting*), esta base deve estar na máquina mestre do NIS e em cada máquina encaminhadora.

Durante a inicialização de uma máquina encaminhadora, cada interface é associada a uma sub-rede (se essa existir) ou rede. Como exemplo, tomemos o caso de uma máquina com 2 interfaces Ethernet que possua endereço de classe B 143.7.1.10 e tenha no arquivo `netmasks` a máscara 255.255.255.0 para a rede 143.7. Essa máquina pode ser definida como máquina encaminhadora usando a interface 1 para comunicar com a sub-rede 143.7.1 e a interface 2 com a sub-rede 143.7.2. Isto é, se for necessária comunicação com uma máquina de endereço 143.7.1.9, a interface 1 será utilizada. Num ambiente UNIX BSD essa associação poderia ser feita da seguinte forma (supondo que o nome da interface 1 seja `ie0` e o nome da interface 2 seja `le0`):

```
ifconfig ie0 inet 143.7.1.10 netmask FFFFFFF0 broadcast 143.7.1.0
ifconfig le0 inet 143.7.2.10 netmask FFFFFFF0 broadcast 143.7.2.0
```

Note que a máquina encaminhadora tem um endereço para cada interface conforme a rede com a qual se comunica diretamente. Nesse caso é 143.7.2.10 para a sub-rede 143.7.2 e 143.7.1.10 para a sub-rede 143.7.1.

Máquinas que desejam comunicar-se com máquinas em outras redes ou sub-redes com as quais não possuem comunicação direta podem fazê-lo usando uma máquina encaminhadora com a qual possuem conexão. Para isso serão necessárias em cada máquina tabelas de encaminhamento que especificam o caminho necessário para alcançar certo endereço. Por exemplo:

```
route add net 143.7.2 143.7.1.10 1
```

Esse comando faz a entrada na tabela de encaminhamento de uma máquina (da sub-rede 143.7.1) que está ligada diretamente pela interface 1 à máquina 143.7.1.10. O comando diz que toda referência a endereços 143.7.2.xx são encaminhados via máquina 143.7.1.10 com métrica 1. A métrica especifica por quantas máquinas precisa passar para chegar ao destino. No caso a métrica é 1 pois 143.7.1.10 possui uma ligação direta com a sub-rede 143.7.2 via interface 2. Para a comunicação realmente existir (ida e volta), cada máquina da sub-rede 143.7.2 deve possuir uma entrada na sua tabela de encaminhamento:


```
route add net 143.7.1 143.7.2.10 1
```

Note que neste caso a máquina encaminhadora é conhecida na sub-rede 143.7.2 pelo endereço 143.7.2.10.

Essas tabelas de encaminhamento podem ser atualizadas e pesquisadas de forma dinâmica. Para isso são utilizados os protocolos RIP, HELLO e EGP [Com88]. No sistema implementado faz-se uso de tabelas estáticas de encaminhamento.

A base de dados que contém o mapeamento nome-endereço é representada, num ambiente UNIX BSD, pelo arquivo `/etc/hosts`. A base de dados `hosts` lista todas as máquinas no domínio local pelo endereço Internet e nome, no seguinte formato:

```
endereço-IP    nome-hospedeiro    apelido    # comentario
```

Para operação correta da rede é essencial que todas as máquinas conheçam o endereço de todas as outras máquinas, e que não haja endereço duplicado. Esse arquivo deve ser atualizado por cada máquina a fim de refletir as mudanças na rede. Se estiverem em operação os softwares NIS ou DNS não é necessária a atualização dos arquivos `hosts` em todas as máquinas. Porém, toda máquina deve possuir no arquivo `hosts` uma entrada para si própria, uma para `localhost` e, se for sem disco, uma para o servidor de arquivos. Um exemplo de arquivo `hosts` seria:

```
192.9.200.1    bailarino
192.9.200.2    ballet      # comentario
127.0.0.1      localhost  localhost
```

O endereço do `localhost` é sempre o mesmo, e é usado por programas para conseguir serviços na mesma máquina em que foram chamados.

A base de dados `networks`, representada pelo arquivo `/etc/networks`, é usada para associar um número de rede a um nome. Cada entrada no arquivo tem a seguinte sintaxe:

```
nome-rede-oficial    numero-rede    apelido    # comentario
```

Um exemplo seria:

```
#
# redes Internet
#
arpanet      10      arpa
# redes locais
loopback     127
eng          193.9.0    # engenharia
cont         193.9.1    # contabilidade
prog         193.9.2    # programacao
```

O `nome-rede`, associado ao `numero-rede` pode ser usado no comando `route`, para entrada na tabela de encaminhamento, ou no arquivo `netmasks` (o que facilita mnemonicamente o gerenciamento das bases). Esse arquivo também pode conter os nomes/endereços das sub-redes locais.

A base de dados `services` contém uma entrada para cada serviço TCP/IP de rede disponível no sistema. Vários programas lêem esse arquivo. Não é necessário mantê-lo. O arquivo `/etc/services` típico é da forma:

```

#
# @(#)services 1.16 90/01/03 SMI
#
echo          7/tcp
echo          7/udp
systat        11/tcp      users
daytime       13/tcp
daytime       13/udp
netstat       15/tcp
ftp           21/tcp
telnet        23/tcp
smtp          25/tcp      mail
time          37/tcp      timserver
time          37/udp      timserver
name          42/udp      nameserver
whois         43/tcp      nicname      # geralmente para sri-nic
domain        53/udp
domain        53/tcp
hostnames     101/tcp     hostname    # geralmente para sri-nic

```

Note-se que vários serviços de rede tais como FTP e Telnet estão listados na primeira coluna. A segunda coluna contém o número de porta do serviço e o protocolo da camada de transporte (TCP ou UDP) que trata disso. As chamadas portas são conceitos usados para, ao invés de pensar em processos com destino final, pensar num conjunto de pontos destino usados pelo TCP ou UDP para demultiplexar datagramas entre os processos a espera de recebê-los [Com88].

5.2 Tratamento dado pela DECNET ao problema de Protocolagem

DECNET é o nome coletivo para produtos de software e hardware que fornecem recursos para vários sistemas operacionais DIGITAL participarem numa rede. DECNET-VAX é a implementação DECNET que permite ao sistema operacional VMS funcionar como um nó de rede. Como uma interface do VMS à rede, o DECNET-VAX suporta os protocolos necessários para comunicação sobre uma rede e as funções necessárias para configurar, controlar e monitorar essa rede.

VAX-PSI (Packetnet System Interface) é um produto de software que permite ao usuário VMS comunicar-se através de uma PSDN (Packet Switch Data Network – rede de comutação de pacotes). Ele implementa as recomendações X.25 e X.29, e os padrões internacionais 7776 e 8208 para prover uma interface do usuário com a PSDN.

O VAX-PSI pode ser utilizado para:

- Ligação de nós DECNET através de uma PSDN, por intermédio de um DLM (Data Link Mapping). Isso permite que um circuito virtual X.25 seja utilizado como um enlace de dados DECNET.
- Comunicação direta através de uma ou mais PSDNs com computadores DIGITAL e não-DIGITAL sobre um circuito virtual X.25.

- Comunicação direta através de uma ou mais PSDNs com um terminal modo caracter conectado a um dispositivo PAD. O PAD pode ser privado ou localizado dentro do PSDN.
- Comunicação via uma ou mais PSDNs, entre terminais no nó local com VAX-PSI e computadores remotos DIGITAL ou não DIGITAL. O VAX-PSI contém um PAD baseado no hospedeiro que provê essa capacidade.
- Comunicação direta com outro computador DIGITAL ou não-DIGITAL sem a intervenção de uma PSDN, contanto que o outro computador implemente os padrões internacionais 7776 e 8208. Neste caso, um computador atua como DTE e o outro como DCE.

Para estabelecer um sistema operacional VMS como parte de uma rede DECNET, deve-se construir e manter uma base de dados de configuração. Esta consiste de registros que descrevem componentes específicos de rede que o sistema em particular requer. Alguns componentes para o DECNET-VAX são: nós, circuitos, linhas, roteamentos, enlaces lógicos e objetos. Alguns componentes para o VAX-PSI são: *X.25 protocol module* e *X.25/X.29 server modules*. O programa NCP (Network Control Program) do DECNET é que permite controlar e monitorar esses componentes.

Nó

Um nó é identificado por um nome e um endereço de 16 bits cujo formato é: *número-da-área.número-do.nó*, onde o número da área assume valores de 0 a 63 (6 bits) e o número do nó, valores de 0 a 1023 (10 bits).

A DECNET suporta roteamento de área, que permite configurar a rede em subredes, chamadas áreas, que consistem em agrupamentos de nós. Suporta o roteamento de pacotes dentro de áreas e um segundo nível de roteamento entre elas. O roteador que efetua o roteamento dentro de uma área é chamado *roteador de nível 1* e o que efetua roteamento entre áreas, bem como dentro de sua própria área, é chamado *roteador de nível 2*, ou *roteador de área*.

Cada roteador de nível 1 mantém informações sobre o estado de todos os nós de sua área. Ele direciona todos os pacotes endereçados a nós fora de sua área ao roteador nível 2 mais próximo. Cada roteador nível 2 mantém informação do caminho de menor custo para as outras áreas da rede, bem como o estado dos nós de sua própria área. Quando um roteador nível 1 recebe um pacote destinado a um nó fora de sua área ele usa um roteamento nível 1 para enviar o pacote ao roteador nível 2 de sua área mais próximo. O roteador de nível 2 passa o pacote ao longo do caminho de menos custo ao roteador de nível 2 mais próximo fora de sua área. O pacote é transmitido ao longo de um caminho de nível 2 até o roteador de nível 2 da área destino. Esse roteador de nível 2 envia os pacotes por roteadores de nível 1 até o nó destino.

A razão básica para a divisão da rede em múltiplas áreas é reduzir o volume de tráfego de encaminhamento que ocorre em uma rede de área única.

Os nós DECNET-VAX *Phase IV*¹ podem ser de 2 tipos:

Roteadores : nível 1, cujo tipo é *routing IV*, e nível 2, cujo tipo é *area*.

¹ *Phase IV* é o nome da versão atual da DECNET

Não-roteadores (nós finais) : Envia e recebem pacotes de outros nós, mas não encaminham pacotes através de si. Esses nós são do tipo *nonrouting IV*.

Mensagens de encaminhamento

Nós encaminhadores adjacentes trocam mensagens de atualização de roteamento. Uma mensagem de atualização de roteamento é um pacote que contém informação sobre o custo (*cost*) e saltos (*hops*) para cada nó na rede. Numa rede com áreas, o roteador nível 1 envia mensagens de atualização de roteamento sobre todos os nós dentro de sua própria área para roteadores adjacentes na área. Roteadores nível 2 enviam mensagens de atualização de roteamento contendo informação de custos e saltos sobre todas as áreas aos roteadores de nível 2 adjacentes.

Sempre que a informação de roteamento muda (por exemplo, quando um circuito cai), novas mensagens de roteamento são enviadas automaticamente. Por exemplo, se alguém alterou o estado de um circuito, tornando o nó remoto inacessível, essa mudança será refletida automaticamente nas mensagens de atualização de roteamento trocadas pelos nós roteadores.

SHOW NET Para ilustrar esses conceitos, vamos analisar o comando **SHOW NETWORK**. Antes de efetuar uma operação específica sobre a rede, pode-se querer checar o estado de disponibilidade de um nó particular na rede. O comando **SHOW NETWORK** emitido por um usuário do VMS (a nível do DCL), mostra a disponibilidade do nó local como membro da rede.

Esse comando mostrará informações sobre outros nós da rede apenas se o nó local for um nó roteador. Se o nó local é nó final (não-roteador), ele não receberá qualquer informação da rede; ao invés disso será mencionado o nó roteador a ele designado. Se o nó é um roteador de área, o comando **SHOW NET** mostrará informação adicional sobre a área.

Constam também das informações fornecidas pelo comando as relações de custo e ligação entre o nó local e os outros nós da rede. As informações apresentadas sobre a rede corrente são as seguintes:

Node : Identifica cada nó disponível da rede pelo seu nome e endereço

Links : Mostra o número de ligações lógicas entre o nó local e cada nó remoto disponível.

Cost : Mostra o custo total de linha do caminho até o nó remoto. O gerente do sistema atribui o custo para cada linha na rede.

Hops : Mostra o número de nós intermediários até o nó destino (incluindo este).

Next hop to node : Mostra a linha física de saída utilizada para alcançar o nó remoto (o nó local é identificado pelo termo **LOCAL**).

Area : Identifica cada área disponível na rede pelo seu número de área. Essa característica é mostrada apenas se o nó local é roteador de área.

Next hop to area : Mostra a linha física de saída utilizada para alcançar a área remota. O nó local é identificado pelo termo **LOCAL**. Também são mostrados o endereço e o nome do próximo salto para a área destino.

Vejamos alguns exemplos. Para um nó roteador de nível 1:

```
$ sh net
```

```
VAX/VMS Network status for local node 46.450 USPFSC on 14-JUN-1991 19:46:14.80
```

The next hop to the nearest area router is node 46.446 FPSP.

Node	Links	Cost	Hops	Next Hop to Node
46.450 USPFSC	0	0	0	(Local) -> 46.450 USPFSC
46.3	0	16	3	X25-FPSP-0-> 46.446 FPSP
46.5 FNALR5	0	16	3	X25-FPSP-0-> 46.446 FPSP
46.6 FNALR6	0	16	3	X25-FPSP-0-> 46.446 FPSP
46.7 FNALRC	0	16	3	X25-FPSP-0-> 46.446 FPSP

(... Informacoes sobre outros nos ...)

Total of 434 nodes.

```
$
```

Para um nó roteador de área:

```
$ sh net
```

```
VAX/VMS Network status for local node 46.446 FPSP on 14-JUN-1991 19:49:42.82
```

Area	Cost	Hops	Next Hop to Area
1	11	5	DSV-0-1 -> 46.435 FNALRA
2	13	7	DSV-0-1 -> 46.435 FNALRA
3	32	11	DSV-0-1 -> 46.435 FNALRA

(... Mais informacoes sobre areas ...)

46	0	0	(Local) -> 46.446 FPSP
48	15	2	DSV-4-0 -> 46.751 BRUC2

Node	Links	Cost	Hops	Next Hop to Node
46.446 FPSP	0	0	0	(Local) -> 46.446 FPSP
46.3	0	6	2	DSV-0-1 -> 46.435 FNALRA
46.5 FNALR5	0	6	2	DSV-0-1 -> 46.435 FNALRA
46.6 FNALR6	0	6	2	DSV-0-1 -> 46.435 FNALRA

(... Mais informacoes sobre nos ...)

Total of 434 nodes.

Algumas características de nó

Além do nome, endereço e tipo do nó, podem-se definir algumas outras características do nó:

Executor : Emitindo comandos de gerenciamento da rede no nó local pode-se efetuar configuração, controle e monitoração de funções que afetam tanto o nó local como outros nós da rede. O nó onde as funções de gerenciamento são realmente efetuadas é chamado de nó *executor*. Usualmente, o nó executor é o nó local. Contudo pode-se entrar no nó local comandos a serem executados em um nó remoto. Para esses comandos, o nó remoto serve como nó executor.

Buffer-size : Determina o máximo tamanho de mensagem que o nó repassa

State : Pode-se controlar o estado operacional do nó local, e assim controlar a ativação de sua participação na rede

Maximum area : Usado em conjunto com o comando `set executor` para limitar o número de áreas que a camada de roteamento do nó executor reconhecerá.

Exemplo de definição de nó usando o NCP:

```
def executor address 40.300 (endereço area 40)
def executor name ze        (nome ze associado ao no' 40.300)
def executor type area      (roteador nível 2)
def executor buffer size 576
def executor state on       (ativo na rede. Para desativar: state off)
def executor max area 46
def node 40.301 name joao
def node 40.302 name antonio
```

Para comandos emitidos por usuário, o nome associado ao nó é utilizado. Por exemplo:

MAIL> send

To: joao::usuario (envia mensagem para usuario, na maquina
joao que esta' em DECNET)

A definição adequada do parâmetro `MAX AREA` é no que está baseado todo o gerenciamento das redes HEPNET e SPAN (como será visto a seguir). A gerência dessas redes afirma que esse procedimento, apesar de não ser claro nem elegante, é um meio efetivo de eliminar o caos que a utilização de uma numeração de área não regulada poderia causar.

Gerenciamento de número de áreas em redes DECNET multi-organizacionais

Áreas HEP e SPAN Como vimos no capítulo 2, a HEPNET, como uma natural extensão de sua arquitetura em forma de estrela de 3 pontas, utiliza 3 áreas DECNET: 41 para o centro oeste, 42 para o centro meio-oeste e 43 para o centro leste. Espera-se que universidades que se conectem a um desses centros utilizem os atributos área/nó do núcleo ao qual se conectam.

Da mesma forma que a HEPNET, cada centro de roteamento da SPAN utiliza seu próprio número de área, e espèra-se que os sistemas remotos utilizem números de nós dentro da área de seu centro de roteamento. Os centros de roteamento da SPAN utilizam os números de área 5 (JPL), 6 (Goddard), 7 (Marshall) e 9 (Johnson).

Redes que colidem – Evolução do problema À medida que uma rede de área ampla torna-se cada vez maior, mais dispersa geograficamente e organizacionalmente fragmentada, o gerenciamento da rede torna-se, obviamente, mais difícil. A expansão das redes HEPNET e SPAN teve que lidar com “complicações externas” decorrentes da necessidade de contato com outras redes DECNET. Esses contatos, considerados indispensáveis, tiveram que ser efetuados com dois tipos de redes: com *redes locais* e com outras *redes de área ampla*.

A complicação do contato com redes locais é basicamente a consequência do estabelecimento, por várias universidades, de redes locais no âmbito de *campus*. Pesquisas de alta energia ou astrofísica requerem freqüentemente o acesso à HEPNET ou SPAN, respectivamente. Se esses sistema já estão conectados à rede local do *campus*, então toda a rede do *campus* ficará ligada à HEPNET ou SPAN. Quando tal conexão ocorre, a rede do *campus* desejará existir como uma área separada. Organizacionalmente, essas LANs são entidades independentes, mas da perspectiva da DECNET, elas são meramente outra área de uma rede de área ampla. O resultado é que tanto HEPNET como SPAN tornaram-se agora uma rede de área ampla contendo não apenas suas próprias áreas núcleo, mas as áreas adicionais das LANs das universidades com as quais estão ligadas.

A segunda “complicação externa” é o resultado de conexão com outras redes de área ampla. No caso da HEPNET, existe necessidade de conexão com a comunidade HEP européia, que é parte de uma rede DECNET baseada em X.25, e que potencialmente conecta muitas redes DECNET acadêmicas e de pesquisa européias. No caso da SPAN, existem ligações com a rede DECNET do “Los Alamos National Laboratory”, e com uma rede de âmbito estadual ligando algumas redes da “Texas University” (TEXnet). Cada rede dessas traz números de área adicionais para o ambiente de rede de área ampla da SPAN.

Assim, a HEPNET e SPAN deparam-se com problemas envolvendo não só seus próprios números de área, mas também os números de área das redes locais e de área ampla com as quais estão ligadas.

A complicação final a uma já complexa e difícil situação foi que a ligação entre a HEPNET e a SPAN tornou-se inevitável. Embora nem a HEPNET nem a SPAN tivessem necessidade de conectar-se, as duas redes descobriram que existia um grande número de redes locais de universidades que necessitavam acesso a ambas as redes. Devido a esse acesso a ambas as redes, essas redes locais acabaram fundindo as duas redes, resultando numa *super-rede* DECNET composta de um número de redes DECNET diferentes, cada uma com sua própria organização e interesse. Enquanto a SPAN, a HEPNET americana, a HEPNET européia, Los Alamos, etc, existem como entidades organizacionais separadas, todas são funcionalmente apenas subconjuntos de uma rede DECNET muito maior. Cada uma dessas redes de área ampla dentro da entidade maior tem suas próprias áreas, e o requisito básico de manter essas áreas únicas. Assim, dentro de uma rede de tal tamanho e diversidade organizacional, o conflito de número de áreas

é certo. A questão torna-se: como o número de áreas deve ser gerenciado numa tal *super-rede*?

Gerenciamento do número de área O gerenciamento do número de área dentro da *super-rede* SPAN/HEPNET/etc seria bem difícil mesmo se fosse meramente um caso de re-embaralhar o uso dos números de área a fim de eliminar qualquer conflito. Infelizmente existe a restrição adicional devida ao pequeno número de áreas que podem ser alocados. Quando foi desenvolvida a DECNET phase IV, tomou-se a decisão de que 63 números de área eram suficientes. No entanto essa quantidade é muito baixa quando se trata da fusão das redes de área ampla. Mesmo se o uso do número de áreas pudesse ser consolidado a se adequar a 63, não existiria a flexibilidade para tratar das necessidades futuras de instalação de redes que não são presentemente parte da *super-rede*. Está claro que o problema não pode ser facilmente remediado, e também que as redes de área ampla dentro da *super-rede* têm interesses pessoais em prevenir qualquer conflito do número de áreas, visto que esses conflitos poderiam prejudicar o roteamento entre áreas dentro da sua específica rede de área ampla. O resultado foi que a SPAN e a HEPNET reconheceram a necessidade de alguma forma de gerenciamento da *super-rede* a fim de tratar da distribuição dos números de área. Os gerentes de ambas as redes concordaram em fornecer uma “frente comum”, tanto para determinar a abordagem adequada para tratar do problema, quanto para assegurar a cooperação de todos os participantes dentro da *super-rede*.

A abordagem escolhida foi investigar mecanismos de “filtragem de área”, que poderia prover maior flexibilidade no tratamento da alocação do número de áreas.

O parâmetro MAX AREA como mecanismo de filtro O filtro com o parâmetro MAX AREA é um mecanismo que tenta criar dois diferentes conjuntos de números de área: um conjunto de números de área baixos, únicos, e um conjunto de números de área altos, reutilizáveis. Pode-se conceitualmente classificar as redes DECNET como servindo a uma de duas funções: redes que servem unicamente a interesses locais e redes que tendem a servir a interesses geográficos e organizacionais mais amplos. Grandes redes de pesquisa como a HEPNET e a SPAN se enquadram na última categoria, sendo redes de área ampla com arquitetura de múltipla-área. Elas certamente devem assegurar que nenhuma de suas áreas conflitem com outras áreas, visto que tal conflito de áreas poderia prejudicar a comunicação entre suas próprias áreas. Adicionalmente, essas redes tendem a servir como “recursos” a um largo espectro de usuários nas comunidades científicas e de pesquisa. Elas beneficiam essas comunidades se evitarem usar qualquer número de áreas que outras, também grandes, redes de pesquisa já estejam utilizando. Assim, redes tais como HEPNET e SPAN possuem o requisito básico para números de área únicos.

Como o filtro MAX AREA funciona O mecanismo de filtro MAX AREA simplesmente usa o parâmetro MAX AREA do nó executor no NCP como filtro entre números de área para redes de área ampla e números de área para redes locais. HEPNET, SPAN, e outras redes de área ampla utilizam números de área menores ou iguais a um dado valor (presentemente 46), e asseguram que não existe conflitos de área entre eles. Nós roteadores de nível 2 dentro dessas áreas deveriam estabelecer o seu parâmetro MAX AREA

Figura 5.2: Utilização de MAX AREA como filtro

de nó executor para 46. Isto resulta que qualquer informação de roteamento de áreas maiores que 46 sejam “abandonadas”, causando efetivamente o desaparecimento dessas áreas para as redes de área ampla. Redes locais ficam então livres para utilizar qualquer número acima de 46. A fim de que um particular nó numa rede local acesse diretamente uma rede de área ampla, por exemplo HEPNET, ele deverá utilizar um número de nó dentro da área HEPNET a que pretende acesso, ao invés do número área/nó dentro de sua própria rede local. Contudo deverá estabelecer o parâmetro MAX AREA para 63, ao invés de 46. Fazendo isso, ele poderá acessar nós da rede local, bem como nós da HEPNET. a fig. 5.2 ilustra o conceito.

Perspectiva da rede acima e abaixo do filtro MAX AREA A perspectiva da rede inteira depende de se ela é vista por nós de número de área acima ou abaixo de MAX AREA. Visto que áreas de baixo número não carregam informação de roteamento sobre qualquer área de alto número, qualquer tentativa por um sistema de baixo número de área de acessar um sistema de alto número de área resultará em uma mensagem imediata:

```
%SYSTEM-F-UNREACHABLE, remote node is not currently reachable
```

Visto da perspectiva dos números de área baixos, os números de área altos simplesmente não existem.

Os números de área alto, contudo, vêm a rede de forma completamente diferente. Seus nós roteadores de nível 2 mantêm informações de roteamento sobre todos os números de área baixos. Qualquer tentativa de acessar sistemas com números de área baixos, contudo, resultará também na mensagem:

```
%SYSTEM-F-UNREACHABLE, remote node is not currently reachable
```

Em contraste com a tentativa de acesso originada nos números de área baixos, essa mensagem não é gerada imediatamente. Ao invés disso ela ocorre após a expiração de um tempo de tipicamente 60 segundos. Basicamente os sistemas de número de área altos vêm os de número de área baixos como alcançáveis, e iniciam a solicitação de uma conexão. O sistema de baixo número de área, contudo, não é capaz de responder, visto que não possui informação de roteamento dos números de área altos. A solicitação de conexão expira e falha.

Em essência, as áreas de alto número vêm as de baixo número como sendo alcançáveis, mas a tentativa de acesso sempre falha. Por exemplo, conforme a fig. 5.2, uma rede de área 57 pode limitar a "visibilidade" das áreas de baixo número colocando no NCP o parâmetro **AREA MAXIMUM HOPS** do nó executor ao mínimo. No exemplo, o roteador de nível 2 da área 57 poderia colocar o seu parâmetro **AREA MAXIMUM HOPS** em 1, um valor que ainda permitiria o acesso à área 42 (uma necessidade se quer comunicar-se com o nó 42.99). Contudo, outras áreas da HEPNET (por exemplo, 41) ficariam mais longe do que um salto, e não poderiam ser vistas por um sistema da área 57.

Como uma observação final, notamos que, enquanto uma área de número alto recebe informação de roteamento de áreas de números baixos, não recebe informações de roteamento de quaisquer outras áreas de número alto conectadas a qualquer das outras áreas de número baixo. Essa informação de roteamento é "filtrada" pelo parâmetro **MAX AREA** nos sistemas de área de baixo número. Assim, podem existir números de áreas altos idênticos, sem criar problemas para as redes locais de cada campus.

Vantagens e desvantagens do parâmetro MAX AREA como filtro O parâmetro **MAX AREA** utilizado como filtro fornece um mecanismo razoavelmente atraente para tratar uma situação tão crítica de números de área. É bastante simples, tecnicamente, de implementar, requerendo apenas o estabelecimento do parâmetro **MAX AREA** no nó executor do NCP. Não envolve parâmetro que têm tendência a requerer ajustes contínuos, como os parâmetros **area max hops** e **area max cost**. Visto que a filtragem real é feita nos nós de rede de área ampla gerenciados de forma centralizada, existe menor possibilidade dos participantes individuais tentarem alterar o acesso para adaptá-lo às suas conveniências. Em resumo, é um mecanismo simples e razoavelmente efetivo que requer uma cooperação entre os participantes afetados.

O filtro sugerido, contudo, não se apresenta como uma solução totalmente "limpa". Uma rede local teria que separar áreas na sua localidade, em contraste com a mais típica facilidade de usar áreas para definir limites. Assim teria-se que arcar com uma sobrecarga adicional de mensagens no uso de roteamento nível 2 para comunicar com seus próprios nós locais que aconteceu estarem em outra área. Existiria também uma sobrecarga adicional de informação de roteamento. Essas desvantagens, contudo, parecem não representarem uma carga insuportável e, na falta de uma solução mais adequada ao problema do número de áreas, tornam-se um preço razoável a ser pago.

Outros componentes DECNET

Outros componentes além do nó são:

Circuitos : Circuitos são caminhos de comunicação virtual entre nós e entre DTEs.

Operam sobre linhas físicas, e são o meio no qual todas as entradas e saídas

ocorrem. Processos DECNET “conversam” sobre circuitos por meio de *conexões lógicas*. Essas conexões carregam uma única corrente de tráfego *full-duplex* entre dois processos a nível de usuário. Podem existir múltiplas conexões lógicas em cada circuito DECNET.

Linhas : Linhas são caminhos físicos de dados entre nós, ou entre DTEs e DCEs (interface de rede X.25)

Objetos : Objetos são processos que recebem solicitação de conexão lógica. Os nomes de objetos DECNET são cadeias alfanuméricas de até 12 caracteres. O software DECNET usa números de objetos como identificadores únicos de objeto. Os números de objetos são únicos através de sistemas, e variam de 1 a 255. Objetos genéricos tais como FAL (*File Access Listener*) e NML (*Network Management Listener*) possuem número diferente de 0, e são reconhecidos através da rede. Normalmente, tarefas definidas por usuário têm o número 0. Contudo, podem ter um número diferente de zero quando provêem um serviço que pode ser utilizado por outros usuários. Os números 128 a 255 são reservados a esse propósito.

Módulos : Módulos são componentes de gerenciamento de rede.

Módulos do VAX-PSI

Módulo X.25-protocol Um sistema operacional com o software VAX-PSI instalado pode funcionar como um DTE (ou DTEs) capaz de enviar ou receber pacotes sobre uma PSDN com a qual o sistema está conectado. Para configurar um DTE local, deve-se estabelecer na base de dados de configuração uma entrada para o módulo X.25-protocol, que identifica a rede com a qual o DTE está conectado. Essa rede representa um caminho para uma PSDN. Deve-se associar o nome de rede com um *perfil* que determina as características da subscrição à PSDN.

Identifica-se o DTE local por um endereço DTE e um nome de rede. Esse DTE deve ser único na rede. Para cada DTE deve-se especificar o estado operacional e o máximo número de circuitos permitidos, e identificar a linha e canais associados para chamadas que saem.

Módulos X.25- e X.29-servers Para tratar chamadas vindas pela PSDN de um DTE ou terminal remotos, configura-se os módulos X.25- e X.29-servers, conhecidos como tratadores de chamada X.25 e X.29 respectivamente. O X.25-server trata chamadas que chegam originadas num DTE remoto. O módulo X.29-server trata chamadas que chegam originadas num terminal remoto.

A configuração da base de dados para esses módulos servidores define os processos a que são “destinadas” as chamadas. Deve-se identificar cada destino por um nome alfanumérico único. Deve-se também nomear o objeto ativado quando um particular processo destino aceita uma chamada que entra, e atribuir prioridades aos processos destinos que podem tratar uma mesma chamada.

Originalmente, pode-se restringir as chamadas que o processo destino irá tratar a qualquer combinação das seguintes:

- Chamadas a específicos subendereços do DTE local

- Chamadas de um específico grupo de usuários
- Chamadas de específicos DTEs remotos
- Chamadas que contém dados de usuário que casam com um valor específico de chamada após uma máscara ser aplicada
- Chamadas redirecionadas de um outro DTE
- Chamadas recebidas de uma específica rede
- Chamadas contendo valores específicos na extensão do endereço chamado após uma máscara ser aplicada.

Exemplo: O nosso endereço RENPAC é 11620020. Se uma chamada especificar 116200202, pode existir um processo destino que atende a essa chamada com subendereço 2. Esse processo faz uma conexão com outra máquina da rede local. Assim, um usuário que especifique o subendereço 2 pode se conectar na outra máquina sem ter que entrar na que está diretamente ligada ao PSDN. A configuração para isso seria (X é o processo que faz conexão com a máquina não ligada à PSDN, e X.COM o arquivo de comandos que define o acesso):

```
NCP> set mod x29-server destination X -
                                subaddress 2 -
                                object      X -
                                priority    127

NCP> set object X -
      number 2 -
      file disco:[diretorio]X.com -
      user decnet -
      password xxxxxx
```

5.3 Informações sobre o conjunto de programas UUCP

UUCP é um nome genérico utilizado para descrever um conjunto de programas que permitem a computadores de sistema operacional UNIX comunicarem-se uns com os outros usando terminais remotos. UUCP denota uma gama de arquivos e utilitários, da qual o programa uucp é apenas uma parte. Esses utilitários variam desde os utilizados para copiar arquivos entre computadores (uucp e uuto) aos usados para *login* e execução remotas (cu e uux).

A primeira versão do UUCP estava disponível com a versão 7 do UNIX em 1978 e era dirigida a comunicação entre sistemas usando linhas telefônicas discadas. Atualmente existem dois tipos principais de UUCP em uso: a versão distribuída com o BSD 4.3, que derivou da versão 7, e a versão conhecida como Honey Danber UUCP, distribuída com o *System V, rel. 3*, onde é oficialmente chamado BNU (Basic Networking Utilities).

O hardware de comunicação atualmente suportado por algumas implementações compreende:

Enlaces diretos : Pode-se criar ligações diretas conectando-se portas seriais entre dois computadores com cabos de até 15 metros.

Linhas telefônicas : utilizando *modems* ou ACU (unidade de chamada automática) para comunicar com outros computadores sobre linhas telefônicas padrões. A ACU disca o telefone solicitado pelo UUCP. O computador que se tenta conectar deve ter um *modem* capaz de responder a chamadas.

Rede : alguns UUCPs também podem comunicar-se sobre redes. Uma vez estando o computador ligado à rede, pode realizar contatos com qualquer outro computador da rede.

Os programas UUCP podem ser divididos em duas categorias: programas de usuário e programas administrativos.

São programas de usuário:

cu : permite a conexão a um computador remoto, possibilitando o *login* em ambos simultaneamente, transferir e executar comandos em qualquer computador sem interromper a ligação inicial.

uucp : permite copiar arquivos de um computador a outro. Cria arquivos de trabalho e de dados, enfileira a tarefa para transferência, e chama o *daemon* **uucico**, que por sua vez tenta o contato com o sistema remoto.

uuto : copia arquivos de um computador a um diretório público em um outro computador. Diversamente do **uucp**, que permite copiar arquivos para qualquer diretório do computador remoto, **uuto** coloca o arquivo num diretório *spool* apropriado e avisa ao usuário para pegá-lo com **uupick**.

uupick : recupera os arquivos colocados no diretório de *spool* pelo **uuto** remoto.

uux : cria arquivos de trabalho, dados, e execução necessários para a execução de comandos no computador remoto. Os arquivos de trabalho contém as mesmas informações que os de trabalho criados pelo **uucp** e **uuto**. Os arquivos de execução contém cadeias de comando para serem executados no computador remoto e uma lista de arquivos de dados. Os arquivos de dados são os necessários para a execução dos comandos.

uustat : mostra o estado das transferências solicitadas (**uucp**, **uuto** ou **uux**). Também provê meios para controlar as transferências enfileiradas.

São programas administrativos:

uulog : Mostra o conteúdo de arquivos de registros. Os arquivos de registros são criados para cada computador remoto com o qual existe comunicação. Os arquivos contém registros para cada uso de **uucp**, **uuto** e **uux**.

uucleanup : limpa o diretório de *spool*. É normalmente executado periodicamente como um *daemon*.²

uutry : testa a capacidade de processamento de chamada e efetua algumas depurações. Invoca o *daemon* do **uucico** para estabelecer o enlace de comunicação entre o seu computador e o remoto especificado.

² *daemon* é um programa que roda como um processo em background

uucheck : verifica a presença de diretórios, programas e arquivos de suporte do UUCP.

Existem dois *daemons* principais no UUCP que já foram mencionados nas descrições acima:

uucico : seleciona o dispositivo usado para a ligação, estabelece a ligação com o computador remoto, efetua a sequência de *login* e verificação de permissão, transfere arquivos de dados e execução, registra resultados, e notifica o usuário (através de *mail*) do fim da transferência. Quando o *daemon* **uucico** local chama o computador remoto, ele “conversa” com o *daemon* **uucico** no computador remoto durante a sessão.

uuxqt : executa solicitação de execução remota. Procura no diretório de *spool* por arquivos de execução (“X.arquivo”) que foi enviado pelo computador remoto. Quando um arquivo “X.arquivo” é encontrado, **uuxqt** abre esse arquivo e pega a lista de arquivo de dados necessários para execução. Então verifica se esses arquivos de dados estão acessíveis e disponíveis. Se esses arquivos estão presentes e acessíveis, **uuxqt** checa o arquivo **permissions** para verificar se tem permissão de executar o comando solicitado.

UUCP para correio

UUCP é usado para transferência de correspondência da seguinte maneira: Um usuário envia mensagem para um endereço da forma **outro-lugar!usuario**. Dois arquivos, um *arquivo de dados* e um *arquivo de execução*, são copiados para o sistema vizinho **outro-lugar**. Após a recepção do *arquivo de execução*, o programa **uuxqt** no sistema vizinho interpreta o conteúdo do arquivo que inclui o comando **rmail** que remete localmente o *arquivo de dados* ao **usuario** receptor mencionado.

A transferência de arquivo via UUCP é limitada a sistemas adjacentes, ou vizinhos, mas é possível enviar mensagem a sistemas não adjacentes via a notação *bang path* (chamada assim pois na leitura em voz alta, a “!” é lida como *bang*). Por exemplo, se temos um nó que conversa com o nó chamado **ibm**, e **ibm** por sua vez conversa com um nó chamado **dec**, e um amigo com nome de usuário **joao** nesta última máquina, pode-se enviar correspondência para **ibm!dec!joao**. Neste caso o programa **rmail** no **ibm** vê como endereço **to: dec!joao**, e enfileira a mensagem de novo no sistema **uucp**, exatamente como se a mensagem tivesse sido montada por alguém no **ibm** para **dec!joao**.

Ao chegar na caixa postal de **joao**, o endereço no **from:** diz **dec!ibm!nome-que-enviou**. Cada sistema coloca seu próprio nome no **from:** à medida que a mensagem passa por ele, de modo que uma réplica possa ser feita.

A característica fundamental do sistema UUCP é ser um sistema que opera em lote. Os comandos **uux** e **uucp** enfileiram um **job**, que é executado algum tempo depois pelo processo **uucico**. A fig. 5.3 ilustra o processo.

Protocolo que uucico suporta

O processo **uucico** original suportava um único protocolo de enlace de dados conhecido como *protocolo “g”*. Este foi desenvolvido para linhas de terminal ou linhas discadas e tipicamente usa pacotes de 64 bytes. A sua capacidade é limitada em torno de 9000

Figura 5.3: Sistema UUCP

bauds, mesmo com linhas de alta velocidade. A versão 4.3 BSD permite dois protocolos adicionais. O **protocol "t"** assume que o canal de comunicação é livre de erro e não é usado nenhum *checksum*. O protocolo é tipicamente utilizado com ligação TCP. O **protocolo "f"** é usado para ligação X.25 e baseado pelo controle de fluxo provido por uma corrente de dados (*data stream*). O protocolo "f" aplica *checksum* apenas no arquivo todo (não em cada pacote) e usa caminho de dados de 7 bits, ao invés dos 8 bits usuais.

Bases de dados de suporte

As bases de suporte são as seguintes:

Devices : contém informações concernentes à localização e velocidade de linha de unidades de chamada automática (ACU), ligações diretas e dispositivos de rede.

Dialers : contém cadeias de caracteres necessárias para negociar com *modems* (ou ACUs) no estabelecimento de conexões com computadores remotos.

Systems : contém informação que o *daemon uucico* e o programa *cu* necessitam para estabelecer conexão com um computador remoto. Contém informações tais como: nome do computador remoto, nome do dispositivo de conexão associado com o computador remoto, quando o computador pode ser acessado, número de telefone, identificação de *login* e senha.

Dialcodes : contém abreviação de código de discagem que pode ser usado no campo número de telefone das entradas no arquivo **systems**.

Permissions : define o nível de acesso que é granjeado a computadores quando estes tentam transferir arquivos ou executar comandos remotamente no seu computador.

Poll : define os computadores que são percorridos pelo seu sistema e quando são percorridos.

Sysfiles : usado para atribuir diferentes ou múltiplos arquivos a serem usados pelo *uucico* e *cu*, tais como arquivos **systems**, **devices** e **dialers**.

Considerações Finais

A partir da análise efetuada neste capítulo foi possível projetar as diversas configurações mencionadas no capítulo 6. Quanto ao TCP/IP na seção 6.4 são mostrados o endereçamento internet utilizado, os domínios envolvidos e as configurações dos vários softwares que o implementam. Quanto ao DECNET na seção 6.1 menciona-se critérios para escolha de endereçamento dos nós e definições para efetivar as conexões necessárias e quanto ao UUCP na seção 6.2 e 6.3 são detalhadas as alterações de softwares e as configurações para conexões entre microcomputadores e o sistema VAX-VMS.

Capítulo 6

Descrição do sistema implementado

Considerações Iniciais

A partir das etapas identificadas no capítulo 3 é apresentada a implementação do sistema de correio eletrônico.

Quanto a etapa de interface de usuário é decidido no capítulo 3 para a implementação do sistema o uso do MAIL-VMS para o VAX-VMS, o ELM para as estações e as interfaces BM-KA9Q e MAIL-UUPC para microcomputadores.

Quanto a etapa de tradução de denominação e determinação do mecanismo de transporte cujas informações básicas são apresentadas no capítulo 4, na seção 6.5 é mostrado o tratamento dado ao PMDF tanto no USPFSC como na FAPESP e no apêndice C o arquivo de configuração do PMDF para o USPFSC. Na seção 6.6 é apresentado como é tratado o sendmail em ambiente com e sem NIS e/ou NFS e nos apêndices D, E e F os arquivos de configuração para as estações floyd, emilia e ethos respectivamente.

Quanto a etapa de protocolagem cujas informações básicas são apresentadas no capítulo 5, na seção 6.1 é mostrado o tratamento dado ao DECNET e o seu relacionamento com o PMDF, na seção 6.2 é apresentada a adaptação do pacote DECUS-UUCP e sua ligação com o PMDF e no apêndice A são mostradas as alterações dos arquivos necessárias, na sessão 6.3 mostra-se como foi acomodado o UUCP e o circuito de arbitramento e no apêndice B figuram os trechos de programas que incorporam a lógica do circuito de arbitramento, e na seção 6.4 o tratamento dado ao TCP/IP, aos vários softwares que o implementam e a sua relação com o PMDF.

Quanto a etapa de transmissão por meio físico são identificados no capítulo 3 as ocorrências: serial, ethernet e X.25.

Na seção 6.7 o funcionamento do sistema é ilustrado através de exemplos mostrando-se as configurações relevantes de cada etapa.

As diversas configurações são apoiadas nos manuais:

- VMS System Management vol 5a: Networking [Dig88b]
- DECSERVER 200 User's Guide [Dig89a]
- DECSERVER 200 Management Guide [Dig89b]

- DECUS UUCP System Manager's Guide [HIP89]
- Fusion Network Software User Manual [Net89b]
- Fusion Network Software Network Administrator's Manual [Net89a]
- Fusion Network Software Programmer's Reference Manual [Net88]
- The KA9Q Internet Software Package [Gar88]
- SunOS System & Network Administration [Sun90a]
- PMDF v3.1 System Manager's Guide [FV89]
- BM User Manual [Tru]
- CLIX TCP/IP User's Guide [Int89]
- Intergraph Network Core (INC) User's Guide [Int90]

6.1 Tratamento dado ao DECNET e seu relacionamento com o PMDF

Para que dois nós DECNET se comuniquem basta que exista um circuito entre eles. Esses circuitos operam sobre linhas físicas. Para configuração Ethernet, o circuito é diretamente relacionado com a linha física. Para configuração X.25 não existe essa correspondência. Todos os circuitos X.25 passam através do módulo *X.25 protocol* que multiplexa os circuitos na linha que conecta o nó à PSDN.

No caso da conexão entre o USPFSC e o MVAXFT sobre Ethernet (fig. 6.1), sendo BNA-0 o nome da interface, os comandos emitidos no NCP tanto no USPFSC como no MVAXFT são:

```
! Estabelece parametros para a linha/circuito BNA-0
NCP> set line BNA-0 state on
NCP> set circuit BNA-0 state on
```

e então associa-se o número dos nós aos nomes que serão conhecidos. Em cada um deles, o endereço próprio e o nome é definido no `set executor`. No USPFSC:

```
NCP> set executor address 46.450 -
           name      USPFSC
NCP> set node 46.1010 name MVAXFT
```

No MVAXFT:

```
NCP> set executor address 46.1010 -
           name MVAXFT
NCP> set node 46.450 name USPFSC
```

Para a conexão entre USPFSC e FPSP sobre X.25 (fig. 6.1), é necessário em cada uma delas que:

Figura 6.1: Conexão entre USPFS e FPSP sobre X.25

- se estabeleça o perfil da subscrição à RENPAC no módulo *X.25 protocol*;
- se indique a linha física usada para a comunicação com a RENPAC;
- se estabeleça o DTE local;
- se estabeleça o circuito usado para a comunicação entre os dois nós sobre a RENPAC.

Os comandos necessários emitidos no NCP são:

```
! Estabelece o perfil da subscricao
set module x25_protocol network renpac profile renpac
! Indica a linha usada para comunicar com a RENPAC
set line DMB-0-0 network RENPAC state on
! Estabelece o DTE local para o no' USPFS
set module x25-protocol -
    dte 11620020 -
    network renpac -
    channels 3-0 -
    line DMB-0-0 -
    state on
```

O USPFS possui quatro canais lógicos de contato com a RENPAC que são numerados de 0 a 3. Um desses canais é usado para o circuito. No nó USPFS:

```
! Estabelece parametros p/ o DLM de saida p/ o no FPSP
set circuit x25-fpsp-out -
    network renpac -
```

```

number 11182594 -
owner executor -
usage outgoing -
state on

```

No nó FPSP:

```

! Estabelece o perfil da subscricao
set module x25_protocol network renpac profile renpac
! Indica a linha usada para comunicar com a RENPAC
set line DSV-0-0 network renpac state on
! Estabelece o DTE local para o nó FPSP
set module x25_protocol -
    dte 11182594 -
    network renpac -
    channels 16-1 -
    line DSV-0-0 -
    state on

```

O nó FPSP possui 16 canais lógicos de contato com a RENPAC que são numerados de 16 a 1. Um desses nós é usado para o circuito:

```

! Estabelece parametros p/ DLM de entrada
set circuit x25-uspfsc-inc -
    network renpac -
    number 11620020 -
    owner executor -
    usage incoming -
    state on

```

Como esses nós são vistos pela HEPNET atualmente

A nossa conexão com a HEPNET é somente através da FAPESP (fig. 6.1) sobre X.25. A FAPESP, por sua vez, se liga com a HEPNET pelo FNALRA (fig. 6.2). Outras instituições como nós usam o mesmo caminho para se conectar à HEPNET. Assim a FAPESP decidiu (atualmente), que todos os nós de instituições brasileiras que pertençam à HEPNET não fossem roteadores de área. Isto foi feito para que quando haja problemas com a conexão com FNALRA (ou nesse mesmo nó) as mensagens de roteamento não busquem caminhos alternativos (que não existem) para conseguir a conexão. Esta busca de caminhos alternativos é uma operação automática do DECNET, para evitar que a queda de um nó ponha outros fora de comunicação numa rede. No nosso caso, o trânsito desses pacotes em X.25 seria completamente supérfluo, pois a única conexão realmente existente com o nó FNALRA é através de FPSP. Desta forma, os nós HEPNET de instituições brasileiras não são roteadores de área (apesar da recomendação nesse sentido em [DeM87]). Desta forma, todos os nós locais (que não pertencem à HEPNET) também têm que estar na área 46, pois de outra forma não poderiam se comunicar com o nó HEPNET da instituição.

O conflito de endereços na área 46 foi solucionado pelo FNALRA, que filtra os números de nó maiores que 750 (fig. 6.2). Assim, os números maiores que 750, na

Figura 6.2: Filtro aplicado ao número de nó

área 46, deste lado do FNALRA, são deixados para nós locais, e se existirem outros conectados ao FNALRA não haverá conflito pois ele os filtra.

No entanto esse procedimento somente é válido no momento, enquanto não existe redundância de conexão com a HEPNET, sendo tudo feito via FPSP e FNALRA. Quando houver outras ligações alternativas, deverá ser possível o estabelecimento de outros roteadores de área, de modo a possibilitar uma maior tolerância a falhas, e o método de utilização de áreas maiores que 46 e o filtro por área será o mais adequado. Porém a idéia de filtro da recomendação ainda existe nesse caso, só que a nível de endereço de nó, dentro da área 46.

Quanto ao roteamento, os nós envolvidos são:

FPSP:

```
NCP> set executor type area
```

USPFSC:

```
NCP> set executor type routing IV
```

MVAXFT:

```
NCP> set executor type nonrouting IV
```

DECNET e PMDF

Canais DECNET são usados para conectar sistemas rodando PMDF-822 que também estão conectados via DECNET. O software do canal DECNET do PMDF usa o protocolo **PhoneNet** em cima de ligações transparentes DECNET.

PhoneNet é um mecanismo de transporte baseado em linha para terminal assíncrono para uso com PMDF. Quando opera no modo *mestre*, é capaz de “disar” sobre linhas telefônicas discadas para estabelecer conexão com um PhoneNet escravo em outra máquina. Quando opera no modo *escravo*, responde a uma solicitação de um mestre remoto para transferência de mensagem.

O protocolo PhoneNet é dividido em dois protocolos separados. Um de nível mais baixo, a nível de enlace, chamado *protocolo de discagem*, provê um ambiente básico de corrente de bytes. É responsável pela aquisição e gerenciamento de conexão, tradução do conjunto de caracteres e controle de fluxo.

O protocolo de nível mais alto, chamado *protocolo de telefone*, é responsável pela semântica de transferência de correspondência. Deve estabelecer o contexto e depois passar as mensagens individuais.

O canal DECNET usa bidirecionalmente os programas mestre e escravo. O programa escravo roda em resposta a uma solicitação DECNET que entra. Para isso deve-se instalar o PMDF como um objeto DECNET conhecido em cada sistema que vai usar o canal DECNET.

```
NCP> set object pmdf -
      number xxx -
      file pmdf_root:[exe]dn_slave.com -
      user conta_pmdf -
      password senha_pmdf -
      account system
```

O número **xxx** pode ser qualquer entre 128 e 255, porém não deve ser usado para qualquer outro propósito, e todos os sistemas devem usar o mesmo número.

O último passo para se estabelecer um canal DECNET é acrescentá-lo nos arquivos de configuração dos PMDFs envolvidos. Os nomes dos canais DECNET têm uma sintaxe rígida: são sempre da forma **dn_no-remoto**, onde **no-remoto** é o nome do sistema a que se conecta. Isso significa que um canal possui diferentes nomes em cada ponta. No caso do nó USPFSC para o nó FPSP, o canal é chamado **dn_uspfsc** no **pmdf.cnf** do FPSP, e **dn_fpsp** no **pmdf.cnf** do USPFSC. Finalmente, para usá-los são necessárias regras de re-escrita nos **pmdf.cnf** que encaminhem para o canal.

6.2 Adaptação do pacote DECUS-UUCP

DECUS UUCP é uma implementação parcial de facilidades UUCP para sistemas VMS. Provê suporte para transferência de *mail* e *news* UUCP, permitindo conexão com redes de correios e notícias UUCP. Não há previsão para transferências de arquivos solicitadas pelo usuário, nem para execução remota de comandos. Ou seja, os comandos **uucp** e **uux** não estão incluídos no software.

A parte central do DECUS UUCP é o **uucico** que, como o **uucico** do UNIX, realiza a transferência de arquivo real de uma máquina para outra usando o protocolo "g". Entre 20 outros programas executáveis e procedimentos de comandos, são providos **uuxqt**, interfaces para correio e notícias, e várias funções de suporte.

O **uucico** do DECUS UUCP é baseado no **gnuucp**, escrito por John Gilmore. O prefixo **gnu** refere-se ao sistema operacional GNU (GNU's Not Unix), um projeto de uma fundação de software livre. O **gnuucp**, por sua vez, é baseado no **uuslave**, um programa encontrado em "bulletin boards" de acesso público.

Com o DECUS UUCP, usuários VMS podem enviar e receber correspondência através do programa MAIL. Para enviar correspondência via UUCP, uma forma especial de endereço **To:** é usada:

To: uucp%"endereço-destino"

Essa função é provida via interface (não documentada no VMS) para protocolo estrangeiro de correspondência do VMS MAIL.

Em termos gerais, para a ativação do DECUS UUCP é necessário:

- criar uma conta para *logins* UUCP
- executar o procedimento de comandos `uucp_systartup`

A conta para *login* UUCP tem um procedimento de entrada cativo. Para *logins* interativos executa a imagem `uucico` e quando essa acaba, o processo faz *logout* na conta. Quando o processamento é em lotes nessa conta, o procedimento de entrada simplesmente acaba de maneira que o processo possa executar qualquer comando especificado para a tarefa em lote.

Quando executado, o `uucp_systartup`:

- associa nomes lógicos aos diretórios corretos
- submete a tarefa em lote `uucico_callout`, que roda o `uucico` no modo *daemon*. O `uucico` procura trabalho de saída (correio ou notícia) para todos os sistemas vizinhos designados no arquivo `systems`. Ele tentará chamar todo sistema vizinho que ou exista algum trabalho de saída para esse sistema, ou foi habilitada a varredura do sistema na entrada do arquivo `systems`. Após isso o *daemon* irá "dormir" até que o intervalo *sleeptime* do arquivo `control` tenha passado, ou até que alguém envie uma *mail* UUCP para saída.
- submete a tarefa em lote `uuxqt_batch` para execução. Essa rodará `uuxqt.exe`, que por sua vez disparará um subprocesso rodando `uxqt_dcl.com`. `uuxqt.exe` procura então por arquivos `X` no diretório `uucp_spool`, se não acha nenhum arquivo ele hiberna.
- submete a tarefa em lote `uucp_cleanup` para ser executada uma hora depois. Esta se resubmeterá a cada hora.

Quando envia-se

MAIL> send

To: uucp%"sistema1!sistema2!...usuario"

o `sistema1` será o nome do sistema vizinho para o qual a correspondência será enviada para ser repassada. Nos nomes de arquivos esse sistema será `eles`, e o sistema de onde é enviado o *mail* será `nos`. `nnnn` nos nomes de arquivos são números inteiros de 4 dígitos.

Quando identifica o `uucp%`, o VMS MAIL ativa a imagem `uucp_mailshr`, que cria três arquivos no `uucp_spool`:

- `B.nosAnnnn` - um arquivo de comando do `uuxqt` para ser enviado ao sistema remoto
- `C.elesAnnnn` - comandos para o programa `uucico` local
- `D.nosAnnnn` - o arquivo de correspondência real, com cabeçalhos RFC822

Quando o **uucico** procura trabalho para **eles**, ou quando decide chamar **eles**, ou quando **eles** chamam **nós**, ele achará o arquivo **C** (também chamado *arquivo de trabalho*). Esse arquivo o instrui a copiar o arquivo **D** para o sistema remoto tal como é, e para copiar o arquivo **B** com o nome trocado para **X.nosAnnnnn**.

Para processar correspondências que chegam, **uucico** acorda **uuxqt** sempre que recebe um arquivo **X**. **uuxqt** procura os arquivos **X** no diretório de *spool*. O arquivo **X** instrui o **uuxqt** sobre os arquivos no diretório a acessar e sobre os comandos a executar no sistema que recebeu, para processar esses arquivos. Para correio, este comando é **rmail**, que entrega a mensagem ao destinatário intencionado.

O arquivo **systems** para conexão com o microcomputador com nome **pcmonic** é da forma:

```
pcmonic - uucp_login g x0x000 Never /
        direct 1200 unused unix usrn timer pwd
```

onde:

pcmonic : é o nome do hospedeiro UUCP remoto (hnome)

- : nome dessa entrada no arquivo **systems**. "-" assume ser o mesmo que hnome. Usado em arquivos de registro e mensagens de depuração

uucp_login : nome de usuário local, que as chamadas que entram irão usar

g : protocolo utilizado (nesta versão é unico, colocado para futuras versões)

Never : nunca chama o hospedeiro (o sistema local é chamado pelo **pcmonic**)

direct, 1200 : deve casar com o nome da porta e velocidade no arquivo contral

unused : não usa número para discar

unix : nome de extensão de descrição de *login* (**uucp.cfg: login.unix**). No caso do **pcmonic**, não utilizado. Seria usado para conexões com máquinas UNIX.

usrnm, timer : cadeias enviadas nas descrições de *login*

O arquivo **control** toma a forma:

```
hostname      uspfsc
debug         000217      ;nivel de depuracao
enbrmtrcv     1           ;usar apenas diretorio uucp_spool
sleeptime     360         ;intervalo grande para acordar so qdo chamado

port          direct none lta2: 1200

spool         uucp_spool:
logfile       uucp_log:uucp.log
logdir        uucp_log:
cfgdir        uucp_cfg:
debugfile     uucico_dbg
sysfile       uucp_cfg:systems
userfile      uucp_cfg:userfile
pubdir        uucp_spool:
```


O mecanismo real usado para “dormir” não é hibernar/acordar (inferido pelo termo “dormir”), mas esperar por um LEF (*Local Event Flag*). Assim, executando o comando **SH SYSTEM** do VMS, ver-se-á que o *daemon* apresenta o estado LEF quando dormindo, e não HIB (hibernando). O *daemon* “acorda” (via uma *mailbox message*, mecanismo de comunicação entre processos no VMS) sempre que alguém envia um correio UUCP para sair.

A linha LTA2: que conecta o VAX ao *pcmonic* é uma linha serial de um DEC-SERVER (servidor de terminais). Sempre que se entra por uma linha dessas é criado um nome de terminal diferente (terminal virtual). Assim, para que o direcionamento do DECUS UUCP funcione, é necessário criar um terminal fixo para essa porta do DEC-SERVER, o que é feito com os comandos:

```
! Cria o dispositivo (device)
CREATE PORT LTA2: /NOLOG /APPLICATION
! Mapeia porto de aplicacao a um porto especifico no servidor
SET PORT LTA2: /APPLICATION /NODE=DS03.VAX /PORT=P3.MONICA
```

PMDF e DECUS UUCP

O PMDF-822 versão 3.1 inclui programas de canal que implementam a facilidade de correio UUCP em cima da implementação DEC/Shell UUCP. Essa implementação é um produto comercial DIGITAL que provê apenas facilidades de transferência de arquivos e execução remota, mas não provê qualquer função do sistema de correio.

O IFQSC não possui o DEC/Shell UUCP, portanto, para fazer com que o PMDF v3.1 trabalhe com o DECUS UUCP são necessárias algumas alterações sugeridas numa *mailing list* de UUCP por Seton R. Droppers. Os arquivos alterados são:

```
uucp_master.pas (PMDf)
uucp_slave.pas (PMDf)
return_uucp.pas (PMDf)
return_uucp.com (PMDf)
master.com (PMDf)
uuxqt_dcl.com (DECUS UUCP)
```

Basicamente corrigem-se os nomes de diretórios e arquivos do DEC/Shell UUCP para DECUS UUCP, adapta-se o mecanismo de varredura do sistema para o utilizado no DECUS UUCP, e faz-se com que o comando remoto utilizado pelo *uuxqt* para entrega de correio seja, ao invés do MAIL com *uucp%* no To:, uma chamada ao canal do PMDF *uucp_slave*.

A título de documentação do sistema implementado, são apresentadas no apêndice A as alterações, mostrando com o utilitário *diff* as diferenças entre os arquivos originais e os arquivos modificados.

São necessários dois ou mais canais no PMDF para comunicar com o UUCP. Um único canal comum é usado por todas as mensagens que chegam, não importando de qual sistema proveio a mensagem. Um canal adicional é necessário para cada sistema conectado via UUCP. O canal das mensagens que entram é apenas escravo, e nunca terá mensagens enfileiradas para ele. O canal das mensagens que saem é apenas mestre.

A entrada no *pmdf.cnf* para o canal das mensagens que chegam é:

```
uucp_gateway uucp slave
uucp_gateway
```

Esse nome de canal é obrigatório.

As entradas para os canais UUCP de saída de mensagens variam dependendo do nome do sistema que o canal conecta. No caso do `pcmonic`:

```
uucp_pcmonic 822 master slave_debug master_debug
uucp_pcmonic
pcmonic.ifqsc.usp.ansp.br pcmonic
```

onde mensagens dirigidas ao nome oficial `pcmonic.ifqsc.usp.ansb.br` são dirigidas a esse canal, sendo o nome UUCP simplesmente `pcmonic`

As regras de re-escrita devem ser estabelecidas de modo a apontar para esse canal de saída usando o nome de hospedeiro oficial do canal:

```
pcmonic.ifqsc.usp.ansp.br $U@pcmonic.ifqsc.usp.ansp.br
pcmonic.uucp             $U@pcmonic.ifqsc.usp.ansp.br
pcmonic                   $U@pcmonic.ifqsc.usp.ansp.br
```

6.3 Acomodação do programa UUPC e Circuito de Arbitramento

O UUPC é um pacote de domínio público obtido de qualquer lugar que arquiva o *newsgroup* da USENET `comp.sources.misc`.

O pacote consiste de dois programas executáveis:

uupc : programa para transferência automática, similar ao `uucico` do UUCP

mail : uma interface de usuário bem rústica, similar ao `mail` do UNIX.

Tipicamente, usuários utilizam o `mail` para compor mensagens que são descarregadas em disco, pois não necessitam ser enviadas imediatamente. Após um lote de mensagens ser preparado, é feita pelo usuário a conexão para transferência de mensagens com um hospedeiro vizinho que possa repassar as mensagens.

O sistema vizinho pode ser um outro micro com UUPC, um sistema UNIX, ou um sistema que fale o protocolo "g" do UUCP.

A sintaxe do comando `uupc` é:

```
uupc -s hospedeiro [-xn]
```

A opção `-x` especifica o nível de mensagens (para depuração da transferência). Os possíveis usos do comando `mail` são:

```
mail -s "assunto" usuario usuario@lugar.remoto.dominio < mensagem
mail -f=caixa-postal
mail
```

O programa `mail` acrescentará um arquivo `signature` se puder achá-lo, e manterá uma cópia da mensagem que saiu em `=mail.sent`.

Uma breve descrição dos arquivos que constituem o pacote é dada a seguir:

dcp.c – contém lógica de alto nível, pseudo-principal do uupc

dcp.h : *header* do uupc

dcpxfer.c : contém a parte de transferência

dcpsys : contém a lógica para conexão

dcpgpkt.c : protocolo de pacote “g”

host.c : programa principal genérico, onde o ambiente da máquina é definido. Usando tanto para o uupc quanto para o mail

host.h : definições que todos os programas necessitam

mllib.c : rotinas de biblioteca usadas apenas pela parte de mail

ulib.c : rotinas de biblioteca usadas apenas pela parte uu

ndir.c : rotinas de pesquisa de diretório padrão *Berkley*

ndir.h : *header* para **ndir.c**

getargs.c : rotinas de biblioteca

lmail define LMAIL, inclui **pcmail**

mail.c : programa de correio (UA)

pcmail.c : programa de correio (MTA)

rmail : define RMAIL, inclui **pcmail**

comm.c : rotinas de interrupção para comunicação serial

comm.h : *header* para **comm.c**

O **pcmail** é a parte de MTA do sistema de correio. Entrega as correspondências locais se existir espaço e o diretório para descarga. E toda a correspondência que sai, vai para um hospedeiro, para ser lá processada.

pcmail/LMAIL A parte de UA de correio é composta do programa **mail.c** e **pcmail.c** compilado com LMAIL. A versão LMAIL do **pcmail** acrescenta todos cabeçalhos RFC822 às mensagens, e copia a mensagem para **=mail.sent**

pcmail/RMAIL O programa uupc contém a versão RMAIL do **pcmail**. Apenas acrescenta os cabeçalhos **From:** e **Received:** para as mensagens que chegam.

A árvore de diretórios usada no pacote é:

```

\usr
\usr\lib
\usr\lib\uucp
\usr\lib\SEQF      - numero de sequencia
\usr\lib\systems   - informacao p/ conexao com hospedeiros
\usr\spool
\usr\spool\mail     - diretorio de correspondencia
\usr\spool\mail\xxx - arq. de corresp. de usuario
\usr\spool\uucp     - diretorio p/ descarga
\usr\spool\uucp\C_YYYYNN.NNN - arquivos de controle de copia
\usr\spool\uucp\D_YYYYNN.NNN - arquivos de dados
\usr\spool\uucp\dcg.log - arquivos de registro
\usr\spool\uucp\X_YYYYNN.NNN - arquivos de controle de execucao
\usr\xxxx           - arquivos de usuarios
\usr\xxxx\.signature - arquivo de assinatura
\usr\xxxx\Mail      - diretorio de correspondencia de usuario
\usr\xxxx\Mail\mail.sent - mensagem enviada
\tmp                - arquivo temporario

```

Para usar o pacote, é necessário estabelecer algumas variáveis de ambiente. O arquivo `setup.bat` faz isso:

```

set MAILBOX=joao
set NAME=Joao Silva
set HOME=\usr\joao
set MAILDIR=c:\usr\spool\mail
set CONFDIR=c:\usr\lib\uucp
set SPOOLDIR=c:\usr\spool\uucp
set PUBDIR=c:\usr\spool\uupublic
set TMPDIR=c:\tmp
set DOMAIN=pcmonic.ifqsc.usp.ansp.br
set NODENAME=pcmonic
set MAILSERV=uspfsc
set DEVICE=com1
set SPEED=1200
SET PARITY=n

```

Para outro usuário utilizar o sistema usa-se:

`login caixa-postal nome sobrenome`

que estabelece algumas variáveis com `login.bat`, listado abaixo:

```

echo off
set NAME=%2 %3
set MAILBOX=%1
set HOME=\usr\%1
echo " "
echo "Logging %2 %3 in as %1..."

```

Configuração local

Para usar o USPFSC como servidor de correspondência, é necessária a entrada no arquivo `systems`:

```
uspfsc Any a DIR 9600 g "" "" local> c sername:--
sername:--sername: uucp_login sword:--sword: uucp00
```

que diz:

- conecta ao USPFSC com velocidade 9600 na porta `a` com o protocolo "g".
- espera nada, envia nada (para dar um tempo)
- espera `local>` (o *prompt* do DECSERVER)
- envia `c` (comando de conexão para o DECSERVER)
- espera `sername:`, se não é recebido envia `<RETURN>`
- espera `sword:`, se não é recebido envia `<RETURN>`
- envia senha

Portanto, essa entrada é consultada ao ser feita conexão para transferência de mensagens via o comando:

```
uupc -s uspfsc [-x10]
```

Na conexão são transferidas tanto as mensagens enfileiradas no USPFSC com destino `pcmonic` quanto as mensagens enfileiradas no `pcmonic` para saída.

Como a conexão é feita de vez em quando, a linha serial da conexão pode ser melhor aproveitada compartilhando-a com outros micros que utilizem o UUPC para transferir mensagens.

Circuito de arbitramento

Para arbitrar esse compartilhamento, foi desenvolvido o circuito da fig. 6.3. que compartilha a linha serial com 4 micros. Na lógica do circuito está implícita uma prioridade entre eles. O primeiro tem prioridade sobre o segundo, o segundo sobre o terceiro e assim por diante.

A lógica do circuito está baseada no aproveitamento dos sinais RTS e CTS da serial. Para usar a linha, o micro pulsa o RTS e lê o CTS para ver se conseguiu a linha. Se não conseguiu, tenta depois.

Analisemos o comportamento dos sinais no circuito: Quando o circuito é ligado, o sinal de *reset* coloca todos os *presets* dos *flip-flops* tipo D em 0, forçando portanto suas saídas para 1. Isto significa que os *buffers* ficam desabilitados, não passando as linhas TX e RX de nenhum dos micros. Também os CTS de todos os micros indicam que ninguém está usando.

Ao terminar o *reset*, os *flip-flops* ficam habilitados (devido à transição do *preset* para 1) a receber as requisições (RTS), representadas pela subida dos sinais de *clock*. Como as entradas D são geradas por um E lógico invertido de todos os Q, que estão

inicialmente em 1, ao ocorrer a transição de *clock* (requisição), a saída do *flip-flop* em que ocorreu irá para 0.

Assim, quando alguém quer transmitir, pulsa o RTS, ou seja, o *clock* do *flip-flop* tipo D, que faz com que a entrada D (em 0) seja copiada para a saída Q. Isto faz com que o valor da próxima entrada D seja 1, o CTS seja ligado, os buffers sejam habilitados, deixando passar o TX e RX do micro correspondente, e os micros de menor prioridade tenham a saída Q forçada para 1.

Após o uso, pulsa o RTS, fazendo com que a entrada D (agora em 1) passe para Q, desabilitando os buffers, sinalizando o CTS, e liberando os de menor prioridade para serem atendidos.

Alteração no pacote

A inclusão do pedido e concessão do uso da linha via os sinais RTS e CTS no UUPC foi feita alterando pequenos trechos dos arquivos:

```
ulib.c
comm.h
comm.asm
```

no software original.

Quando é tentada a transferência de mensagem via a ativação do uupc, são percorridos os vários estados necessários no programa *dcp.c* através da implementação:

```
while (TRUE)
{
    switch (state) {
        case estado1:
            state=funcao_de_estado();
            ...
        case estadon:
            state=funcao_de_estado();
    }
}
```

onde cada função de estado pode retornar um valor que o coloque em outro estado. Tanto no modo mestre como no escravo, os estados iniciais estabelecem os parâmetros da linha serial, e verificam se conseguiram estabelecer com sucesso, testando o retorno da função *openline()* definida no arquivo *ulib.c*. Essa função, por sua vez, usa a função *dtr_on()*, definida no arquivo *comm.asm*, que coloca ativo o sinal DTR para avisar que o terminal está pronto para transmitir. Nos estados finais de ambos os modos, é usada a função *closeline()*, definida no arquivo *ulib.c*, que por sua vez usa a função *dtr_off()* definida no arquivo *comm.asm* para avisar que o terminal não está pronto, o que no caso do *modem* faz com que “ponha o fone no gancho”.

Assim, alterou-se a função *dtr_on()* a fim de pulsar o RTS e ler o CTS para ver se conseguiu a linha, e a função *dtr_off()* para pulsar o RTS para liberar a linha.

Com essa alteração, o *dtr_on()* retornará um valor que será usado para o retorno da *openline()*, que vai permitir que se passe para estados que continuam a transferência ou desistem.

A liberação final da linha é feita através dos estados finais que usam *closeline()*, que liberará a linha com *dtr_off()*.

Figura 6.4: Registradores de controle e estado da 8250

A programação da alteração está baseada no registrador de controle de *modem* e no registrador de estado de *modem*, mostrados na fig. 6.4.

Os trechos dos programas que sofreram alteração são apresentados no apêndice B.

6.4 Tratamento dado ao conjunto de protocolos TCP/IP

Endereços Internet das máquinas locais

Entre as instituições brasileiras que têm registro na Internet estão a FAPESP e a USP. O endereço de registro para ambas é de classe B: 143.108 para a FAPESP e 143.107 para a USP. Dentro da USP convencionou-se criarem-se sub-redes, dentro desse endereço de classe B, para cada unidade da USP. Assim, estão destinados a cada unidade os endereços 143.107.unidade.xxx. Portanto, para o IFQSC, tem-se 143.107.54.xxx, e para o ICMSC 143.107.55.xxx. Uma outra convenção brasileira foi definir que entidades menores, sem registro no NIC (Network Information Center), ficassem sob o endereço de registro da FAPESP. Assim escolheu-se, localmente, para o UEPAESC o endereço 143.108.100.2, que logicamente seria a máquina 2 dentro da sub-rede 100 do endereço de classe B 143.108 da FAPESP.

Portanto, os endereços TCP/IP das máquinas locais ficam como ilustrado na fig. 6.5.

Domínios Internet envolvidos

Ainda sujeito a mudanças, os domínios Internet envolvidos na implementação estão todos sob a organização lógica **ansp.br** (*academic network* São Paulo, do Brasil). A fig. 6.6 mostra os domínios Internet dentro do ambiente organizacional do sistema implementado.

Configuração para acesso de 143.107.54 a 143.107.55 e 143.108.100.2

É necessário mascarar os bits do endereço de classe B 143.107 para que seja possível distinguir as subredes 54 e 55.

Figura 6.5: Endereços TCP/IP

Figura 6.6: Domínios Internet

Normalmente a máscara de classe B seria 255.255.0.0 (2 octetos para rede e dois para hospedeiro). Para a existência das sub-redes é necessária a máscara 255.255.255.0 (3 octetos para rede e um para hospedeiro).

Conforme ilustrado na fig. 6.5, tanto a estação de trabalho **floyd** quanto a **ethos** possuem interfaces **slip** para comunicação entre as sub-redes 54 e 55, e Ethernet para comunicação dentro de cada sub-rede. Para definir a interface **slip** entre a **ethos** e a **floyd** é necessário definir em cada uma delas:

```
sliplogin endereco-local endereco-da-ponta < qual-porta &
```

que dispare um processo que fica em *background* atendendo solicitações via o protocolo IP em cima da serial [Rom90]

Assim, as máquinas de cada sub-rede têm acesso à outra sub-rede via a máquina da sub-rede que possui conexão **slip** com uma máquina da outra sub-rede. Portanto, o comando, que precisa ser incluído em cada máquina que queira acessar a outra sub-rede é:

```
route add net outra-subrede maquina-com-slip 1
```

onde **outra-subrede** é o nome lógico da rede com a qual se deseja conexão, e a métrica 1 significa que se possui conexão direta com a **maquina-com-slip**, e esta, por sua vez, possui conexão direta com a sub-rede. Portanto, a distância entre a máquina e a sub-rede é 1.

Para a conexão **floyd-Embrapa** é necessário definir outro processo **sliplogin** na **floyd** usando outra linha serial, e definir no software KA9Q na Embrapa o uso da serial. Nesse caso não são necessárias máscaras, pois o endereço 143.108 já é outra rede para os endereços 143.107, e vice-versa.

A sintaxe dessas configurações para cada máquina nos vários softwares TCP/IP é mostrada a seguir:

Estações SUN-IFQSC

A floyd e a emilia (representada na fig. 6.5 como outras SUNs) compartilham várias bases administrativas do TCP/IP através do NIS. Portanto, são comuns os arquivos listados abaixo. Arquivo hosts:

```
#
# A entrada seguinte e' exigida para teste de retorno de eco (loopback)
127.0.0.1 localhost localhost
#
143.107.54.10      floyd  mailhost
143.107.54.1       uspfsc
143.107.54.2       lepton
143.107.54.4       neumann
143.107.54.6       meson1
143.107.54.7       meson2
143.107.54.8       meson3
143.107.54.11      lieip3
143.107.54.12      dfcmip
143.107.54.15      iris.ifqsc.usp.ansp.br iris
143.107.54.20      xtjan
143.107.55.13      icmsc.usp.ansp.br ethos
143.108.100.2      uepaesc.embrapa.ansp.br
```

Arquivo netmasks:

```
# Rede          mascara
143.107         255.255.255.0
```

Arquivo networks:

```
#
loopback 127
floyd-ifqsc      143.107.54
ethos-icmsc      143.107.55
```

Na floyd é necessário definir:

```
sliplogin floyd ethos < /dev/ttyb &
```

para a conexão com a ethos, e

```
sliplogin floyd uepaesc < /dev/ttya &
```

para conexão com a Embrapa.

Para a emilia acessar a sub-rede do ICMSU é necessário o comando:

```
route add net ethos-icmsc floyd 1
```

e se quiser acessar a Embrapa:

```
route add host uepaesc floyd 1
```

Estações SUN-ICMSC

Da mesma forma a **ethos** e as outras SUNs podem compartilhar os arquivos do NIS:
Arquivo **hosts**:

```
#
127.0.0.1 localhost
#
143.107.55.13      ethos loghost
143.107.54.10      floyd
143.107.54.1       uspfsc
143.107.54.4       neumann
143.107.54.5       emilia
143.107.54.6       meson1
143.107.54.7       meson2
143.107.54.8       meson3
143.107.54.11      lieip3
143.107.54.12      dfcmip
#      TRIBO do LDCC
143.107.55.7       tupy
143.107.55.1       sparc2a
143.107.55.2       sparc2b
```

Arquivo **netmasks**:

```
#
# Rede mascara
# para a rede de classe B 143.107 parte de hospedeiro e' xxx.xxx
# poder ter sub-redes 143.107.55 e 143.107.54 para poder rotear por
# outra interface, no caso o slip, ao inves de ethernet para os hosts locais
143.107              255.255.255.0
```

Arquivo **networks**:

```
loopback 127
# rede do icmsc: ethos-icmsc, rede do ifqsc: floyd-ifqsc
floyd-ifqsc 143.107.54
ethos-icmsc 143.107.55
```

Para conexão com a **floyd**, é preciso definir na **ethos**:

```
sliplogin ethos floyd < /dev/ttya &
```

Para as outras SUNs acessarem a sub-rede do IFQSC, é necessário emitir em cada uma delas o comando:

```
route add net floyd-ifqsc ethos 1
```

Estações Interpro

A implementação TCP/IP nas Interpro é chamada CLIX TCP/IP. A base de dados atualizada via o comando CLH (*clearing house*) é que armazena os nomes dos nós e seus endereços, tanto TCP/IP quanto XNS e Ethernet.

O programa **namex** usa essa base para construir o **/etc/hosts**. Isso permite que o programa de inicialização do TCP/IP (**dodini**) e programas de aplicação TCP/IP

acessem o `/etc/hosts` ao invés da base CLH. Esse programa é automaticamente executado 8 minutos após o *boot* e diariamente às 12:10 horas. Pode ser emitido manualmente (pelo *super-user*):

```
# /usr/ip32/inc/namex [-u -v]
```

`-v` indica modo *verbose*, e `-u` força a construir o `/etc/hosts`, mesmo que para o sistema ele já esteja atual.

O programa `dodini` na hora do *boot* lê o `/etc/hosts` para achar o endereço do nó local, e inicializa os protocolos TCP/IP. Também permite estabelecer manualmente máscara de sub-rede e carregar a tabela estática de roteamento. A sintaxe é:

```
# /etc/dodini [-a] [-r nome-tabela] [-s mascara]
```

onde:

`-a` opção usada na hora do *boot*

`-r nome-tabela` O conteúdo dessa tabela é da forma:

```
NETWORK: 143.107.55.0    GATEWAY: 143.107.54.10
```

para dizer que a rede `ethos-icmsc` é acessada via `floyd`

`-s 255.255.255.0` para poder distinguir as sub-redes 54 e 55 no endereço 143.107 de classe B.

Se a tabela é `r_tbl` pode-se colocar o comando

```
/etc/dodini -r r_tbl -s 255.255.255.0
```

no arquivo `/etc/rc2.b/S02D0DINI` que é ativado no *boot*.

O comando `netaddr` mostra o endereço do nó local e a máscara ativa.

O CLH não compartilha automaticamente informação de endereços Internet com nós não Intergraph. Para fazer entrada dos nós é necessário:

- Acessar o CLH numa Interpro
- Criar um objeto para nó não Intergraph. O nome do objeto torna-se o nome do nó
- Apagar a propriedade `XNS_ADDRESS` e os valores colocados automaticamente nos novos objetos
- Criar uma propriedade `TCP_ADDRESS` e colocar para seu valor o endereço Internet do nó.

KA9Q

É um pacote público de TCP/IP para PC-DOS, obtido na base SIMTEL20.ARPA, no diretório PD: <MSDOS.PACKET>. Provê facilidades para FTP, Telnet (simples, sem opção de terminais) e SMTP.

É constituído principalmente dos programas NET e BM, onde BM é a interface de usuário para correio. A configuração do NET está no arquivo `autoexec.net`, que deve ficar na raiz. Essa configuração é constituída por um conjunto de comandos que são executados quando o NET é executado. Ou seja, esses comandos podem ser emitidos manualmente dentro do NET:

NET> comando

A configuração para o KA9Q da Embrapa deve estabelecer a interface serial, e a do microcomputador xtjan a interface Ethernet. A configuração de cada um deles é mostrada a seguir. Para o xtjan:

```
#
# Esta entrada indica ao NET.EXE o nome de sua maquina.
#
hostname xtjan
#
#-----
#
# Esta contem o endereco Internet
ip address [143.107.55.20]
#
#-----
#
# O comando attach indica ao NET.EXE a interface que sera usada para o
# TCP/IP. Pode incluir portas seriais, controladores Ethernet, ou cartoes
# dedicados de pacotes
#
# A sintaxe e:
# attach <hw tipo> <endereco de E/S> <vetor> <modo> <rotulo> <tambuff>
#          <mtu> [<velocidade>]
#
# Interface 'lan' ethernet (packet driver)
# para isso nao esquecer de instalar 'ne1000 0x60'
#
attach packet 0x60 lan 5 1500
#
# *****
#
# Esta entrada diz ao programa para rotear todos os pacotes para a
# interface lan descrita no attach
route add default lan
#
# *****
#
#-----
#
# ttl (Time To Live) e' o numero maximo de hops que um pacote pode passar
```

```

# antes de ser eliminado. Este comando previne um loop infinito inadvertido
# com os pacotes na rede
#
ip ttl 16
#
#-----
#
# O mms (maximum segment size) e' o maior tamanho de uma transmissao simples
# que se deseja receber. O valor 216 forca os pacotes a serem no maximo 256
# (contando o overhead)
#
tcp mss 216
#
#-----
#
# O parametro window estabelece o numero maximo de bytes que podem
# ficar pendentes antes do seu sistema esperar um acknowledge.
# Se window e' duas vezes maior que mss, exitiram 2 pacotes ativos no
# canal em qualquer momento. Altos valores de window possibilitam uma
# melhora na taxa de transmissao, mas apresentam problemas. Recomenda-se
# manter mss <= window <= 2*mss.
#
tcp window 432
#
#-----
#
# Esta entrada abre net.log no diretorio \spool e registra a atividade
# do server do seu sistema. Se voce nao quiser um registro, elimine esta
# linha.
#
log net.log
#
#-----
#
# Cada servico que a maquina ira fornecer deve ser ligado antes
# de ficar ativo. As seguintes entradas ligam todos os servicos.
# Para desligar o servico usa-se o comando 'stop' dentro do NET
# ou retira-se a linha correspondente neste arquivo
#
start smtp
start ftp
start echo
start discard
start telnet
# o retransmissor de correio e a floyd
smtp gateway floyd
smtp trace 2
smtp kick

    Para o uepaesc:

# Nome da maquina
hostname uepaesc
#

```

```

#-----
#
# endereco Internet
ip address [143.108.100.2]
#
#-----
#
# definicao da interface
# COM1 ajustada para ligacao slip a 9600 com outro computador
#
attach asy 0x3f8 4 slip sl0 8092 576 9600
#
# *****
#
# roteamento.
# As entradas abaixo dizem para rotear os pacotes para o endereco
# 143.107.54.10 para a unidade sl0 descrita no comando attach. E todo
# pacote para qualquer outro endereco rotear via a interface sl0 para o
# endereco 143.107.54.10 com metrica 1.
#
route add 143.107.54.10 sl0
route add default sl0 143.107.54.10 1
#
# *****
#
#-----
#
# ttl (Time To Live)
#
ip ttl 16
#
#-----
#
# mms (maximum segment size)
#
tcp mss 216
#
#-----
#
# window
#
tcp window 432
#
#-----
#
# net.log no diretorio \spool
#
log net.log
#
#-----
#
# servicos
#

```



```

start smtp
start ftp
start echo
start discard
start telnet
# o retransmissor de correio e a floyd
smtp gateway floyd
smtp trace 2
smtp kick

```

Note-se que o retransmissor de correio para ambas é a **floyd**. Ou seja, qualquer mensagem que sai é endereçada para a **floyd**, para lá passar pelo despachador **sendmail** antes de ser enviada.

O domínio e nome de usuário, bem como a caixa postal e o diretório que mantém a cópia das mensagens enviadas são indicados no arquivo **bm.rc**, de configuração do **BM**. **bm.rc** da Embrapa

```

host uepaesc.embrapa.ansp.br
user sin
fullname Setor de Informatica da UEPAESC
mbox c:\mbox
record c:\outmail
edit c:\bin\vi

```

bm.rc do **xtjan**

```

host xtjan.ifqsc.usp.ansp.br
user jan
fullname Jan Slaets
mbox d:\recebidos
record d:\enviados
edit \util\ne

```

Para manter listas de postagem existe o arquivo **alias** de formato:

```
sinonimo destino1 ... destinoN
```

Fusion

Atualmente está-se usando no VAX-VMS a implementação TCP/IP feita pela **FUSION**.

Para a configuração aqui desejada são necessárias entradas nos arquivos **machine.db**, para entrada dos nós Internet, e **net.db** para definição da máscara e informação de roteamento. Esses arquivos estão mostrados a seguir. O arquivo **machine.db**:

```

machine name:xna address:ipa address:comment
uspfsc::143.107.54.1:VAX 6420
lepton::143.107.54.2:VAX 11/780
neumann::143.107.54.4:Sun SPARC 1
emilia::143.107.54.5:Sun SPARC 1
meson1::143.107.54.6:Sparc Station 1 +
meson2::143.107.54.7:Sparc Station 1 +
meson3::143.107.54.8:Sparc Station 1 +
floyd::143.107.54.10:Sun SPARC GX

```

```

lieip3::143.107.54.11:Intergraph IP32
dfcmip::143.107.54.12:Intergraph IP220
iris::143.107.54.15:Silicon Ghraphics Station - Unix
pcjan::143.107.54.20:XT Jan NE1000
ethos::143.107.55.13:Sun SPARC 1 (ICMSC)
uepaesc::143.108.100.2:PC UEPAESC slip
uspfsc.ifqsc.usp.ansp.br::143.107.54.1:VAX 6420
lepton.ifqsc.usp.ansp.br::143.107.54.2:VAX 11/780
neumann.ifqsc.usp.ansp.br::143.107.54.4:Sun SPARC 1
emilia.ifqsc.usp.ansp.br::143.107.54.5:Sun SPARC 1
meson1.ifqsc.usp.ansp.br::143.107.54.6:Sparc Station 1 +
meson2.ifqsc.usp.ansp.br::143.107.54.7:Sparc Station 1 +
meson3.ifqsc.usp.ansp.br::143.107.54.8:Sparc Station 1 +
floyd.ifqsc.usp.ansp.br::143.107.54.10:Sun SPARC GX
lieip3.ifqsc.usp.ansp.br::143.107.54.11:Intergraph IP32
dfcmip.ifqsc.usp.ansp.br::143.107.54.12:Intergraph IP220
iris.ifqsc.usp.ansp.br::143.107.54.15:Silicon Ghraphics Station - Unix
pcjan.ifqsc.usp.ansp.br::143.107.54.20:XT Jan NE1000
ethos.icmsc.usp.ansp.br::143.107.55.13:Sun SPARC 1 (ICMSC)
icmsc.usp.ansp.br::143.107.55.13:Sun SPARC 1 (ICMSC)
uepaesc.embrapa.ansp.br::143.108.100.2:PC UEPAESC slip

```

E o arquivo `net.db`:

```

name:exists:state:route:hops:delay:maxpkt:options:comment
veva0:1:nrc$veva0:[ETHER]aa000400c2b9:1:10000:1024:DEVICE=ETA0:
veva0:1:nrc$veva0:[INET]143.107.54.1:1:10000:1024:GWY;SUBNET=0.0.255.0:
veva0:1:nrc$veva0:[INET]->143.107.54.10:1:10000:1024:GWY:
veva0:1:nrc$veva0:[INET]143.107.55.0->143.107.54.10:1:10000:1024:GWY;
SUBNET=0.0.255.0:gateway floyd para ethos

```

Note-se que o equivalente do arquivo `hosts` é o arquivo `machine.db` e o sinônimo da máquina é implementado com outra entrada para o mesmo endereço. Note-se também que no arquivo `net.db` a interface Ethernet é `veva0` e a máscara é `0.0.255.0`, que equivale à máscara aqui escolhida `255.255.255.0`.

Fusion e PMDF

Para interfacear com o PMDF, é necessário definir na base `service.db` do Fusion que o programa `FTCP_slave` do PMDF atuará como servidor da porta 25 TCP. O servidor interpretará solicitações SMTP que chegarem conforme o documento RFC821 [Pos82]. A entrada na base `service.db` é da forma:

```
smtpd:1:tcp:25:::pmdf_root\[exe]ftcp_slave:PMDF SMTP server
```

A definição do canal Fusion TCP/IP no `pmdf.cnf` é da forma:

```
ftcp_local single_sys smtp
FTCP_DAEMON
```

Os nomes dos sistemas acessíveis via TCP/IP podem ser acrescentados no bloco do canal ou ser dirigidos pelas regras de reescrita ao canal `ftcp_local`.

Se necessitar-se passar mensagens para sistemas entroncamentos acessados via SMTP, basta criar canais adicionais para cada sistema retransmissor. Os nomes desses canais devem começar com `ftcp_` e o bloco de canal tem a forma:

```
ftcp_entroncamento smtp daemon router
nome_do_entroncamento
```

As chaves `daemon router` dizem ao `FTCP_master` não abrir conexão direta com o primeiro nome do sistema do endereço de envelope mas, ao invés, abrir a conexão com o nome de hospedeiro oficial do canal.

No caso, se a `floyd` fosse entroncamento de alto nível também, a entrada do bloco de canal seria:

```
ftcp_floyd smtp daemon router
floyd
```

o que faria que mensagens dirigidas ao canal não fossem entregues para o `To:`, e sim passadas para a `floyd`. Lá o `sendmail` da `floyd` despacharia as mensagens.

6.5 Adaptação do despachador PMDF

PMDF-Fapesp

O USPFSFSC é o entroncamento local para o mundo, e a Fapesp o entroncamento nacional. Portanto as mensagens que chegam na Fapesp endereçadas a `uepaesc`, `icmsc` e `ifqsc` devem ser destinadas ao USPFSFSC para então serem entregues. Assim são necessárias as seguintes regras no `pmdf.cnf` da Fapesp:

```
ifqsc.usp.ansp.br      $UQ$D
.ifqsc.usp.ansp.br     $U%$H$D@ifqsc.usp.ansp.br
icmsc.usp.ansp.br      $U%$D@ifqsc.usp.ansp.br
.icmsc.usp.ansp.br     $U%$H$D@ifqsc.usp.ansp.br
uepaesc.embrapa.ansp.br $U%$D@ifqsc.usp.ansp.br
.
.
.
on_uspfsc queue        dec$batch
ifqsc.usp.ansp.br      fsp.fapesp.br
```

que dizem que as mensagens destinadas aos domínios `ifqsc`, `icmsc` e `uepaesc` devem ser enviadas pelo canal cujo nome oficial de hospedeiro é `ifqsc.usp.ansp.br`, que é o canal DECNET/PhoneNet da Fapesp com o USPFSFSC.

PMDF-USPFSFSC

De maneira geral, as regras no `pmdf.cnf` do USPFSFSC estão definidas de forma a:

- ajustar denominações incompletas, tanto para aceitar mensagens quanto para tornar as denominações dos `From:` completamente especificadas (lembrando que as regras são usadas tanto no `from:` quanto no `To:`)
- enfileirar as mensagens conforme a denominação destino nos canais adequados
- tudo o que não conseguir entregar localmente é passado para a Fapesp.

O `pmdf.cnf` local é mostrado no apêndice C.

Observe-se que:

- na regra “coringa” “.”, o `@fppsp.fapesp.ansp.br` é colocado como encaminhamento fonte no endereço que sai. Por causa disso, mensagens que saem por esse caminho sempre voltam (via `reply`) pelo PMDF da Fapesp
- o nome de hospedeiro do USPFSC quando usa o canal DECNET é `uspfsc.hepnet`
- os sinônimos para sistemas alcançados via canal PSI são o número de DTE dos nós. Para enviar *mail* a um DTE de uma PSDN normalmente usa-se o PSI através do VMS-MAIL, com sintaxe, por exemplo:

```
MAIL> send
```

```
To: psi%11620020::usuario
```

onde 11620020 é o número do DTE. Assim no PMDF os sistemas alcançados via PSI-mail utilizam para endereço de envelope esses sinônimos (que são os números de DTE dos nós).

- todo nome de hospedeiro incompleto, como por exemplo `emilia`, é posto na sua forma completa: `emilia.ifqsc.usp.ansp.br`. Isto é feito para que o `From:` saia da forma completa, de modo a ser possível o `reply`. No entanto isto também implica que para a entrada da mensagem é tentada a conexão SMTP com o endereço do nó `emilia.ifqsc.usp.ansp.br`. Portanto tem que existir no arquivo `machine.db` (equivalente ao `hosts`) uma entrada

```
emilia.ifqsc.usp.ansp.br    numero-endereco-emilia
```

para que se possa resolver o endereço `emilia.ifqsc.usp.ansp.br` e efetuar a conexão SMTP.

As mensagens para `.hepnet` não passam pelo PMDF da Fapesp, mas usam a conexão via Fapesp de modo transparente, através da DECNET. Isto também acontece com os nós alcançáveis via *Rede de Pacotes*, que utilizam o canal PSI.

Isto implica que os `From:` que saem daqui não passam pelas regras da Fapesp, e os despachadores destino não são conhecidos. O caminho percorrido não é através da denominação de domínios Internet. Isto faz com que as mensagens que chegam no nó destino não possam voltar com `reply`.

Esse é o caso do CITHEX, da HEPNET. Se uma mensagem é com destino a `cithe1.caltech.edu`, esta mensagem vai pelo PMDF da Fapesp, que encaminha a outros nós e despachadores até alcançar o destino. A mensagem assim enviada é recebida no CITHEX por um despachador que recebe mensagens Internet. Se o usuário fizer `reply` na mensagem, esta volta sem problemas. Se a mensagem for enviada para `cithex.hepnet`, vai diretamente via conexão DECNET para o nó CITHEX, e é recebida por outro despachador que recebe mensagens DECNET. Apesar desse caminho ser mais rápido e direto, as mensagens, por motivos exteriores, não serão passíveis de `reply`.

Outro caso de comportamento diverso é o tratamento de `From:` vindo do `pcmonic`. O PMDF local, como recebe a mensagem de um canal UUCP, mantém a exclamação

no **From:**. Apesar de conseguir colocar o nome completo com domínio no **From:**, a exclamação e inversão do UUCP se mantém: **pcmonic...br!usuario**. Se sai diretamente esse **From:** torna a mensagem não passível de **reply**. Se sai pela regra “.”, o PMDF da Fapesp faz a inversão, pois não recebe por canal UUCP, e a mensagem torna-se passível de **reply**.

A solução para esse caso seria criar um sinônimo de caixa postal local para usuários que recebem mensagens no **pcmonic**. Exemplo, no arquivo **alias**.

```
monipc: monica@pcmonic
```

e criar uma entrada na **reverse.dat**:

```
monica@pcmonic  monipc@uspfsc
```

O que acontece quando a mensagem é recebida por canal UUCP e tem destino externo é que o **From: pcmonic...br!monica** é transformado pela entrada na **reverse.dat** numa denominação que não usa o canal UUCP. (Note-se que para ser efetuada tanto a conversão do **alias** quanto do **reverse.dat**) o nome resultante deve possuir um canal para ser enviado, senão não é efetuada a conversão. Com esse procedimento, o PMDF assegura que existe um canal para enviar o nome, ou seja, não é um nome absurdo). Portanto a inversão é feita localmente, e o **From:** sai sendo uma caixa postal do nó USPFSC (local para o PMDF). Esse **From:** é passível de **reply**. Quando a mensagem volta para **monipc@uspfsc**, a entrada no **alias** assegura que será recebida no **pcmonic**.

No arquivo **alias** estão os sinônimos de caixas-postais enfileiradas no canal local. Ou seja, endereçada a USPFSC.

Se quiser-se identificar o domínio **ifqsc.usp.ansp.br** com o entroncamento local USPFSC, pode-se enviar mensagens com **From: caixa-postal@ifqsc.usp.ansp.br** e ter no **alias** um sinônimo para cada uma, contendo o nome do usuário e a máquina onde quer receber. Note-se que o usuário pode ter conta no USPFSC e receber mensagens lá, quando usa o nome de sua conta, ou em outra máquina, quando nas mensagens usa o nome **caixa-postal**. Para isso, são necessárias as entradas, no **reverse.dat**:

```
nome@maquina          caixa-postal@uspfsc
```

e no **alias**.

```
caixa-postal@uspfsc  nome@maquina
```

Para sinônimos de caixas postais não enfileiradas no canal local, pode-se utilizar o recurso de chave de pesquisa **\$(texto)** para a **general.dat**. Por exemplo, a entrada abaixo no arquivo **pmdf.cnf**:

```
icmsc.usp.ansp.br      $U%$( $U)@canal
```

No caso, está-se usando o próprio nome de usuário (caixa-postal) como chave de pesquisa na base **general.dat**:

```
antonio                ethos.icmsc.usp.ansp.br
```

Esse exemplo faria com que mensagens dirigidas a `antonio@icmsc.usp.ansp.br` fossem recebidas na **ethos** pelo usuário **antonio**.

Uma utilidade para esse recurso seria manter centralizado no próprio USPFSC caixas postais dos domínios que o USPFSC é entroncamento. Apesar da viabilidade, isso não é conveniente, pois implicaria em tabelas muito grandes, muita dependência dos domínios com o USPFSC, e o trabalho de manutenção das tabelas árduo e incessante. A administração de caixas-postais distribuídas pelos domínios é mais coerente. Esse assunto será retomado na próxima seção.

6.6 Acomodação do despachador Sendmail

Sendmail em ambiente com NIS e NFS

Num ambiente de máquinas compartilhando as bases administrativas com NIS, usuários com entrada na `passwd` distribuída pelo NIS têm acesso a qualquer máquina desse ambiente.

As correspondências de usuários chegam no diretório `/var/spool/mail`. Ou seja, em cada máquina, existe um arquivo `nome-usuario` nesse diretório que contém as mensagens a serem entregues ao usuário. Se existir uma máquina servidora de correspondência, ou seja, que exporta o diretório `/var/spool/mail` para máquinas clientes, um usuário terá acesso à mesma correspondência, não importando a máquina em que entrou.

Com o NFS (Network File Service) isto pode ser feito incluindo uma entrada do diretório `/var/spool/mail` no arquivo `/etc/exports` da máquina servidora:

```
$ cat /etc/exports
```

```
.
.
.
/var/spool/mail
```

```
.
.
.
```

executando os comandos:

```
$ exportfs -a
$ nfsd 8 &
```

incluindo uma entrada na `fstab` de cada máquina cliente que deseja importar o diretório:

```
$ cat /etc/fstab
```

```
.
.
.
estacao-servidora:/var/spool/mail /var/spool/mail nfs,rw
.
.
.
```

e executando, nessas estações clientes:

```
$ mount -a
```

Ainda pode-se escolher uma das máquinas como hospedeira de correio, identificada nos arquivos de configuração como **mailhost**. A candidata é a máquina que possui interfaces com mais de uma rede. Isso é efetuado dando o sinônimo **mailhost** a essa máquina no arquivo **hosts**, instalando nessa máquina a configuração **sendmail.main.cf** e nas outras **sendmail.subsidiary.cf** com a entrada de retransmissor para o **mailhost**.

```
$ cat sendmail.subsidiary.cf
```

```
.
.
.
```

```
DRmailhost
```

```
.
.
.
```

No arquivo **sendmail.cf** do **mailhost** (**sendmail.main.cf**), pode-se identificar a máquina com o domínio do NIS.

```
$ cat sendmail.cf
```

```
.
.
.
```

```
Dj$m
```

```
.
.
.
```

e incluir uma regra para retirar o nome de máquina do **From:** para todas as mensagens que saem através do retransmissor do domínio.

Dessa forma, mensagens para usuários desse ambiente (NIS+NFS+Sendmail) que chegam ao **mailhost** dirigidas ao domínio, ou que chegam a **maquina.dominio**, são acessadas da mesma forma pelo usuário, não importando a máquina em que entre.

Note-se o conceito, no Sendmail, de máquina hospedeira de correio (**mailhost**) e retransmissor de domínio. A máquina hospedeira de domínio é usada pelo Sendmail das máquinas subsidiárias para repasse de mensagens que não consegue resolver localmente. E o retransmissor de domínio é usado pelo Sendmail da máquina "principal", hospedeira de correio, para repasse de mensagens que não conseguiu resolver no domínio. Isto é feito com os comandos seguintes. No **sendmail.cf** das subsidiárias:

```
DRmailhost
```

e no da principal:

```
DRretransmissor
```

No sistema implementado, a máquina principal do domínio **ifqsc.usp.ansp.br** é a **floyd**, e o retransmissor é o **USPFSC**. No caso da **ethos** ser a máquina principal do domínio **icmsc.usp.ansp.br** o retransmissor é a **floyd**.

Figura 6.7: Uso de sinônimos pelo Sendmail

Sendmail sem NFS ou NIS

Quando o ambiente não compartilha o diretório `/var/spool/mail`, mensagens que chegam a usuário numa determinada máquina não serão as mesmas se ele entrar em outra máquina. Portanto, um usuário num ambiente com NIS sem NFS pode desejar receber as mensagens numa máquina determinada. Para isso usa-se o recurso dos arquivos de sinônimos. O uso desses arquivos pelo Sendmail é mostrado na fig. 6.7.

A utilização desses arquivos deve ser feita com bastante cuidado, para evitar *loops* e viagens adicionais. Podem ser usados para listas de postagem, direcionar mensagens a um determinado programa, incluir arquivos de sinônimos atualizados por outrem e redirecionar mensagens para outra pessoa quando em viagem.

A colocação ideal do sinônimo que faz o usuário receber suas mensagens numa determinada máquina é na base **aliases** do NIS. Ideal pois o NIS já faz a inversão quando a mensagem sai do domínio. Por exemplo, o seguinte comando no arquivo **aliases**:

```
maria    jose@maquina1
```

faz com que mensagens que chegam a **maria@dominio** sejam passadas para **jose@maquina1**. Se **jose** manda mensagens de **maquina1** para um destino fora do domínio, o Sendmail pesquisa o arquivo **aliases** e coloca no **From:** a parte esquerda mais o domínio: **maria@dominio**. Assim, mensagens respondidas a esse **From:** chegarão ao usuário mesmo que ele mude de máquina (e, claro, atualize o **aliases**).

Para um ambiente sem NIS, deve-se identificar o **mailhost** com o domínio, colocar o sinônimo na `/etc/aliases` do **mailhost** e criar uma regra para que mensagens que saem do domínio tenham o **From:** **usuario@dominio**.

Sem NIS, o domínio deve ser colocado na entrada **Dm** do **sendmail.cf**:

```
Dmdominio
```


O lugar ideal para ativar a regra é no despachador usado para o retransmissor. A regra **S** do despachador deve ter:

```
R$*<$+.LOCAL>$*      $$!<Q$m>$3
```

Outra forma de identificar a máquina com o domínio é colocar como sinônimo da máquina no **hosts** o domínio, e dizer no **sendmail.cf** que a máquina é identificada apenas com o conteúdo do **hosts**:

Dj\$w

Arquivos **sendmail.cf** locais

floyd

O **sendmail.cf** da **floyd** é um arquivo de configuração de máquina principal, pois ela é hospedeira de correio do domínio **ifqsc.usp.ansp.br**. O retransmissor é o **USPFSC**, e o despachador utilizado é o **relay**. Existem regras para conseguir enviar para **icmsc** e **uepaesc** e regra para permitir entregar mensagens para máquinas no domínio que não estejam no **hosts**, como o **MVAXFT**. Ela é identificada, no **Sendmail**, como **maquina.dominio**. O **sendmail.cf** da **floyd** é mostrado no apêndice D.

emilia

O **sendmail.cf** da **emilia** é um exemplo de arquivo de configuração de máquina subsidiária. O arquivo é uma simplificação do **sendmail.main.cf** do qual várias regras foram subtraídas. Existe a regra:

```
R$*<@LOCAL>$*      $1<@m>$2      hospedeiro=entroncamento de dominio
```

não existente no **sendmail.cf** do principal.

As regras que na principal são para resolver hospedeiro local para **LOCAL** são usadas para apagar redundâncias de informação local.

Também existe a regra:

```
R$*<@$$y>$*      $1<@2.LOCAL>$3
```

para tornar os nomes locais (do **hosts**) completamente qualificados.

A **emilia** usa para repassar as mensagens o **mailhost** do domínio, ou seja, a **floyd**. É identificada como **maquina.dominio**. O único despachador usado é o **ether**.

O **sendmail.cf** da **emilia** é mostrado no apêndice E.

ethos

O **sendmail.cf** da **ethos** é um arquivo de configuração de máquina principal, pois será a hospedeira de correio do domínio **icmsc.usp.ansp.br**. A máquina retransmissora é a **floyd**, via o despachador **relay** (apenas nome, efetua TCP/IP em cima de linha serial com SLIP, de modo transparente ao **Sendmail**). Ela é identificada com o domínio no **Sendmail** ou com **Dj\$w** se existir um sinônimo do domínio no **hosts** ou com **Dj\$m**. Para os exemplos mencionados adiante escolheu-se a opção **Dj\$w**. Se o **NIS** estiver instalado,

a entrada **Dm** é dispensada e a regra no despachador para retirar o nome da máquina também.

Note-se que os despachadores **relay** e **ether** usam o mesmo veículo. O recurso da criação do despachador **relay** é poder criar regras que serão usadas apenas quando sai para o retransmissor.

O **sendmail.cf** da **ethos** é mostrado no apêndice F.

dfcmip

É necessária a definição de domínio, pois não existe o conceito de domínio do NIS no sistema. A sintaxe do **sendmail.cf** é:

```
DDifqsc.usp.ansp.br
CDifqsc.usp.ansp.br estacao
```

A máquina retransmissora é a **floyd**, e o despachador é o **ether**. Além do despachador **ether** também existe o **uucp** para entrega de *mail* para a **lieip3** em cima de XNS. Essa transferência em cima de XNS dos arquivos UUCP é feita pelo programa **uxmailr** que possui entrada na **crontab** e é disparado pelo **cron** a determinados intervalos de tempo.

Os nomes dos hospedeiros com UUCP ficam na classe V. A classe S é definida para permitir também hospedeiros que não são objetos de CLH. Portanto, no conjunto de regras 0 os hospedeiros dentro da classe S são enviados via despachador **ether**, e os da classe V via **uucp**.

6.7 Exemplos do funcionamento do Sistema

Serão mostrados a seguir alguns exemplos de correspondência ilustrando as várias etapas de correio pelas quais passam.

USPFSC→cithex.hepnet

Observe-se o que acontece com a correspondência que sai do nó USPFSC destinada ao usuário **usp** da máquina **cithex.hepnet** (fig. 6.8)

O usuário emite o comando

```
MAIL> send
To: in%"usp@cithex.hepnet"
```

O **in%** ativa o PMDF que percorre o **pmdf.cnf** tentando encontrar uma regra que indique o canal pelo qual a mensagem deve ser enviada. É encontrada a regra **.hepnet**, que indica o canal **decnet_channel**. O bloco do canal diz que o USPFSC sai nesse canal com o nome **USPFSC.HEPNET**.

O PMDF então passa a incumbência ao DECNET, que deve passar a mensagem do nó 46.450 (USPFSC) ao nó 41.60 (CITHEX). Como o nó local está na área 46 e o destino na área 41, o nó local, que é apenas roteador *dentro* da área 46, tenta achar um circuito com um nó da área 46 que seja roteador de área. Esse nó roteador de área é o FPSP, e o circuito usado é o **X25-FPSP-Outgoing** em cima de X.25. A conexão do USPFSC com o FPSP é tentada através de RENPAC. No FPSP, o circuito **X25-USPFSC-Incoming** permite a conexão. O FPSP, sendo roteador de área, mas não

Figura 6.8: Correspondência do USPFS a `usp@cithex.hepnet`

Figura 6.9: Correspondência de pcmonic a jif

possuindo nenhum circuito adjacente para a área 41, passa para um roteador de área adjacente. Esse roteador adjacente é o FNALRA, que possui um circuito com um roteador de área da área 41. Esse roteador (2) da área 41, não possuindo um circuito adjacente para o nó 41.60, passa para um roteador *dentro* da área 41 (1), com quem tem um circuito adjacente. Esse roteador passa para outro roteador (1) até que seja alcançado um que tenha circuito adjacente com 41.60. Após estabelecido o circuito, a mensagem é entregue. No From: da mensagem aparecerá `usuario@USPFSC.HEPNET`.

pcmonic → jif

A fig. 6.9 fornece uma ilustração do caminho percorrido por uma mensagem enviada por `monica@pcmonic.ifqsc.usp.ansp.br` com destino a `usp@jif.berkeley.edu`.

O usuário compõe a mensagem com o mail do UUCP que cria os arquivos B, C e D no diretório `spool` com instruções de cópia (C) e entrega (B) da mensagem (D) para o hospedeiro remoto.

Os arquivos ficam no diretório até que alguém tente a conexão com o hospedeiro

que processará as mensagens, emitindo o comando:

```
uupc -suspisc
```

Quando esse comando é emitido, busca no arquivo **systems** instruções para conexão com o USPFSC: porta de saída, velocidade, conta remota e senha.

Com essas instruções o UUPC tenta entrar na conta **uucp_login** com a senha **uucp00** no terminal fixo **LTA2**: que está associado à porta 3 do DECSERVER. Se consegue entrar, o procedimento **login.com** da conta ativa o **uucico** que, conversando via protocolo "g" *sliding window* com o UUPC transfere os arquivos para o diretório **spool** do USPFSC.

Esses arquivos são mensagens (D) e instruções de comandos sobre as mensagens (X). O *daemon* **uuxqt**, verificando a presença do arquivo X, irá executar a instrução **rmail** contida no arquivo. A execução da instrução **rmail** no **uuxqt**, ao invés de chamar **mail.uucp%**, irá chamar **uucp_slave** (pela alteração de Seton Droppers), que passará a mensagem para dentro do PMDF se existir o canal **uucp_gateway** no **pmdf.cnf**. O PMDF então tenta achar uma regra para decidir por qual canal enviar a mensagem. A única regra que bate é a *coringa* "." que manda enviar pelo canal **dn_fpsp** com roteamento fonte **@fpsp.fapesp.ansp.br** no endereço.

Esse canal faz a conexão DECNET em cima de X.25 com a Fapesp, ativando o objeto **dn_slave** definido na base de configuração DECNET via comandos do NCP.

O **dn_slave** passa a mensagem para o PMDF que percorrerá o **pmdf.cnf** para achar um canal que envie para o **jif.berkeley.edu**. Após determinado o canal, a mensagem passará por outras máquinas e despachadores até alcançar o destino.

jif→pcmonic

A fig. 6.10 ilustra o caminho que a mensagem que o usuário **usp@jif.berkeley.edu** envia ao usuário **monica@pcmonic.ifqsc.usp.ansp.br**.

A mensagem sai do **jif**, passa por várias máquinas e despachadores até alcançar a Fapesp. Na Fapesp, o PMDF determina que o canal para enviar para **.ifqsc.usp.ansp.br** é o **dn_uspfsc**. Este canal ativa sobre a conexão DECNET-X.25 com o USPFSC o objeto **dn_slave**. O **dn_slave** passa a mensagem para o PMDF, que determinará que o canal para enviar para **pcmonic.ifqsc.usp.ansp.br** será **uucp_pcmonic**, e o nome destino será **pcmonic**. Essa determinação faz com que o **uucp_master** esvazie a fila do PMDF correspondente ao **uucp_pcmonic** e coloque a mensagem no diretório **spool** através da criação dos arquivos B, C de D. O *daemon* **uucico**, quando esgota o **sleeptime**, verifica no arquivo **systems** que o **pcmonic** não é para ser chamado, e dorme novamente. Quando o **pcmonic** chama o USPFSC através do comando:

```
uupc -suspisc
```

o **uucico** e o **uupc** interagem, e a mensagem é transferida para o diretório **spool** do **pcmonic** na forma dos arquivos D e X. Quando o usuário designado (**monica**) quiser ler a mensagem, o **mail** do UUPC a mostrará.

xtjan→Embrapa

A fig. 6.11 ilustra as etapas que uma mensagem de **monica@xtjan.ifqsc.usp.ansp.br** percorre com destino ao usuário **sin@uepaesc.embrapa.ansp.br**.

Figura 6.10: Correspondência do jif ao pcmonic

Figura 6.11: Correspondência do xtjan ao uepaesc

O usuário compõe a mensagem com o **BM** do **KA9Q** que busca no **BM.RC** o endereço completo, com domínio, do usuário, bem como o seu nome, a fim de compor o **From:**.

Quando é ativado o **NET** do **KA9Q**, o **smtp kick** do **autoexec.net** faz com que ele tente a conexão com o entroncamento **SMTP:floyd**. Para ativar a conexão, verifica no **hosts.net** o endereço da **floyd**. A linha:

```
route add default lan
```

no **autoexec.net** diz que, para qualquer endereço, deve-se usar a interface **lan**. A linha **attach**, por sua vez, diz que a interface **lan** usa o **packet driver** **NE1000** instalado em **0x60**.

Assim, a conexão com a **floyd** é executada via Ethernet. Na **floyd**, a conexão **SMTP** ativa o **Sendmail**, que percorre o **sendmail.cf** para determinar o despachador que enviará a mensagem. No conjunto de regras 0 existe a regra específica para **uepaesc** que diz para enviar via o despachador **ether**. Esse despachador tenta conexão **SMTP** com o **uepaesc** buscando seu endereço no arquivo **hosts**. O endereço no **hosts** diz que ele está em outra rede. O processo **sliplogin** diz que o nó com esse endereço pode ser acessado via **SLIP** (**TCP/IP** em cima de serial) pela porta **ttya**. Como o **NET** no **uepaesc** pode não estar ativo o tempo todo, o **Sendmail**, após não conseguir resposta **SMTP**, coloca a mensagem no **/usr/spool/mqueue** até que numa conexão posterior possa transferi-la.

Quando alguém ativa o **NET** no **uepaesc**, a linha

```
smtp kick
```

faz com que se tente conexão **SMTP** para transferência de mensagens. Para isso usa o entroncamento **SMTP:floyd** cujo endereço se encontra no **hosts.net**. Pelas linhas **route** e **attach**, o **NET** determina que para qualquer endereço será usada a interface **s10** que usa o protocolo **SLIP** em cima de **COM1:**. Após essa determinação é feita a conexão **SMTP** com a **floyd** via o processo **sliplogin** e o **Sendmail**, e a mensagem é transferida para o diretório do usuário **sin**. Quando esse usuário desejar ler, ativará o **BM** do **KA9Q**.

Embrapa(uepaesc)→Embrapa(ntiaa)

Na fig. 6.12 mostram-se as etapas percorridas por uma mensagem emitida pelo usuário **sin@uepaesc.embrapa.ansp.br** com destino ao usuário **pan@ntiaa.embrapa.ansp.br**.

O usuário compõe a mensagem através do **BM**. O **From:** desta, através do **BM.RC** conterá nomes e endereços completos. Ativando o **NET**, é tentada a conexão **SMTP** com o entroncamento **floyd** via a interface **s10**, que usará **SLIP** na **COM1:**. A conexão **SLIP** é permitida na **floyd** pelo processo **sliplogin**, que lê a serial **ttya**. O pedido **SMTP** ativa o **Sendmail**, que tenta achar um despachador para enviar a mensagem. Como não consegue resolver o endereço no domínio, passa para o retransmissor **USPFSC**. Assim, é tentada a conexão **SMTP** com o **USPFSC** via Ethernet. No **USPFSC**, a implementação **TCP/IP** da **Fusion** consulta o arquivo **service.db** e verifica que o processo **ftcp_slave** é que deverá atender a conexão **SMTP**. Esse processo passa a mensagem para o **PMDF**, que determina, pela regra “.”, que a mensagem deve ser enviada pelo canal **dn_fpsp**

Figura 6.12: Correspondência de uepaesc a ntiaa

com roteamento fonte `@fppsp.fapesp.ansp.br`. Esse canal, pela conexão DECNET-X.25 com a Fapesp, ativa o objeto `dn_slave` que passa a mensagem para o PMDF.

O PMDF do nó FPSP, através da regra `ntiaa` no seu `pmdf.cnf`, determina que a mensagem deve ser enviada por um canal PhoneNet em cima do PAD da Rempac. Após a transferência ser efetuada, o usuário `pan` no `ntiaa` poderá ler a mensagem com alguma interface para correio.

Outras SUNs→MVAXFT

A fig. 6.13 ilustra as etapas por onde passa uma mensagem que um usuário (*alguem*), nas SUN do IFQSC que não são roteadoras, representadas na figura pela *emilia*, envia para o usuário `joao@mvaxft.ifqsc.usp.ansp.br`.

O Sendmail da *emilia* verifica, por uma regra `S3`, que chama a `S6`, que o endereço tem o domínio do NIS, e coloca `LOCAL`, retirando o domínio. Quando passa para `S0` determinar o despachador é chamada a `S9` para recolocar o domínio, pois MVAXFT não estava em `hosts`, o que implica que a mensagem não pode ser entregue localmente, devendo ser repassada ao `mailhost`. O `mailhost` indicado no `hosts` distribuído pelo NIS é a *floyd*. É para ela que a mensagem é repassada. Na *floyd*, a regra `S6` foi alterada para só pôr `LOCAL` em endereços com domínio cujos hospedeiros estão no `hosts`. Isso evita que depois, na `S0`, passe pela regra que diz que nunca ouviu o hospedeiro (*never heard of host*) no domínio `ifqsc.usp.ansp.br`. (Pois o MFAXFT logicamente está no domínio `ifqsc.usp.ansp.br`, mas não está no `hosts` pois é apenas parte da DECNET, assim a comunicação com ele é só via USPFSO.) E portanto que a mensagem possa ser enviada para o hospedeiro retransmissor (USPFSO) via Ethernet, para que lá seja resolvida.

A conexão SMTP *floyd*-USPFSO ativa o `ftcp_slave` que passa a mensagem para o PMDF. O PMDF então determina pela regra `.ifqsc.usp.ansp.br` que deve ser usado o `decnet_channel` para enviar a mensagem. A mensagem é transmitida via DECNET ao MVAXFT, e o usuário `joao`, via VMS-MAIL, verá no `From:` que a mensagem veio de `alguem@emilia.ifqsc.usp.ansp.br`

USPFSO→ethos

A fig. 6.14 ilustra as etapas por onde passa a mensagem que um usuário no USPFSO envia para o usuário `joao@icmsc.usp.ansp.br`.

O usuário, enviando a mensagem com

```
MAIL> send
```

```
To: in%"joao@icmsc.usp.ansp.br"
```

ativará o PMDF, que irá determinar que o canal `ftcp_daemon` é aquele no qual as mensagens a `icmsc.usp.ansp.br` devem ser enviadas. O canal vai tentar conexão SMTP com o endereço especificado para `icmsc.usp.ansp.br` no arquivo `machine.db`. Esse endereço diz que está na mesma rede de classe B, mas a entrada no arquivo `net.db` diz para aplicar uma máscara no terceiro octeto. Com isso, o endereço fica em outra sub-rede (55), e deverá ser acessado via a máquina 143.107.54.10, que é a *floyd*. Os processos `sliplogin` na *floyd* e na *ethos* permitem uma conexão SMTP entre USPFSO e *ethos* de modo transparente (Ethernet-serial).

Figura 6.13: Correspondência da emilia para o MVAXFT

Figura 6.14: Correspondência do USPFC à ethos

(Se a regra no `pmdf.cnf` dirigisse a mensagem ao canal `ftcp_floyd` com as chaves `daemon router`, a mensagem seria passada para o Sendmail da `floyd` e esse determinaria com a regra específica para o `icmsc` o despachador para enviar a mensagem. O despachador `ether` usaria a conexão SLIP e o Sendmail falaria SMTP com a `ethos`, a fim de entregar a mensagem.)

A conexão SMTP ativa o Sendmail na `ethos`. No `sendmail.cf` a máquina local está identificada com as entradas `ethos` e `icmsc.usp.ansp.br` no `hosts` para o endereço 143.107.55.13. Se existir o NIS com domínio `.icmsc.usp.ansp.br` a entrada `Dm` é dispensada.

A seguir é analisado como o endereço `joao@icmsc.usp.ansp.br` passa pelas regras do `sendmail.cf`:

Na S3:

```
R$+@$+ $:$1<@2>
```

produzindo `joao<@icmsc.usp.ansp.br>` e continua na S3 por causa do `$:`. Casa com:

```
R$+<@+> $@>6$1<@2>
```

que chama a regra 6 passando `$1<@2>` como `joao<@icmsc.usp.ansp.br>` para a entrada da S6. A S6 retorna a entrada sem alterar. A S3 termina nessa regra, por causa do `$@` antes da chamada de S6 (`$>6`).

Na S0 casa com:

```
R$*<$*=w>$* $1<$2LOCAL>$4
```

produzindo `joao<@LOCAL>`. Casa com:

```
R$*<@LOCAL> $@>30$1
```

que chama a regra 30 com a entrada `$1=joao`. A regra 30 repete o processo, pois chama S3 de depois S0.

Na S0 (da segunda vez), casa com:

```
R$+ $#local $:$1
```

Assim, a mensagem é resolvida para o despachador local. Se existisse a entrada no `aliases`:

```
joao: joao@bororo
```

O processo se repetiria para a entrada `joao@bororo`. Pelas regras:

S0

```
$*<@%y>$* $1<@2.LOCAL>$3
```

```
$*<@y.LOCAL>$* #ether $@2 $:$1<@2>$3
```

determina-se que `bororo` está no `hosts` (pelo `%y`) e acrescentaria-se o `.LOCAL` no endereço. Isso faz com que o endereço produzido case com a regra de baixo, e produza:

```
#ether $@bororo $:joao<@bororo>
```

que envia a mensagem com destino `joao<@bororo>` pelo despachador `ether`, que fará a conexão com `bororo` para a entrega. A regra `S21` do despachador não altera `joao<@bororo>` e a `S4` retira os `<>`, tornando o endereço `joao@bororo`.

Se a mensagem for respondida e o NIS estiver rodando, o `From:` já sairá:

```
joao@icmsec.usp.ansp.br
```

Se não tiver NIS, a regra

```
R$*<@$.LOCAL>$*  $0$1<@m>$3
```

no `S31` do despachador `relay` fará com que seja retirado o nome da máquina do `From:` e seja colocado o domínio apenas.

Um último comentário sobre o Sendmail: a sua ativação se faz com:

```
/usr/lib/sendmail -bd -q1h
```

Sempre que se alterar o `sendmail.cf`, deve-se matar o processo acima e redisparar, para que as alterações sejam efetivadas.

Capítulo 7

Conclusão

Este trabalho demonstra a viabilidade de se implementar correio eletrônico em sistemas heterogêneos, tanto no que tange a sistemas operacionais quanto a protocolos e meios físicos. Assim foi analisada e concretizada a troca de correio eletrônico entre sistemas MS-DOS, VMS, UNIX-SYSTEM V (CLIX) e UNIX-BSD (SunOS) em meios físicos tais como Ethernet, X.25, e linhas seriais (com e sem *modem*).

Os sistemas implementados permitiram, na maioria dos casos, não somente o uso de correio eletrônico entre as máquinas locais, mas também *login* remoto e transferência de arquivo.

O estudo realizado permitiu travar contato com os vários protocolos de comunicação e atestar suas peculiaridades, opções e implicações da configuração de suas bases de dados, conforme as necessidades específicas do nó.

Foi possível constatar o conceito de camadas nos softwares de comunicação, e atestar que as conversas ocorrem em camadas do mesmo nível.

As informações sobre redes contidas no capítulo 2 bem como as referências contidas nesse trabalho e na minidissertação [Tra90] permitem situar no contexto atual de redes tanto para fins informativos quanto para fins de estudo.

A forma como foi tratada o problema da heterogeneidade para o sistema implementado, através da identificação de etapas, dá diretrizes para uma abordagem sistemática de outras aplicações em ambientes heterogêneos.

O estudo e também a utilização dos despachadores contidos nesse trabalho verificam o conceito de entroncamento bem como dão exemplo de como *softwares* de alto nível podem tratar o problema de heterogeneidade.

A coletânea dos conjuntos de protocolos utilizados no ambiente heterogêneo em que se implantou o sistema também mostra aspectos relevantes que têm que ser levantados para a implementação de aplicações nesse tipo de ambientes.

A documentação existente nesta dissertação, que atesta o funcionamento da rede presentemente implementada, poderá também auxiliar a implementação e expansão de sistemas semelhantes em outros locais, tendo em vista a carência tanto de recursos humanos especializados nesta área quanto a falta de documentações e descrições de implementações concretas em sistemas heterogêneos na literatura técnica (livros) e na documentação dos vários softwares utilizados (manuais).

O sistema tal como descrito nesta dissertação é estático. Como futuros trabalhos podemos citar a implementação usando busca de **nome-endereco** dinâmicos, bem como

roteamento para os protocolos TCP/IP.

Apêndice A

Arquivos modificados para adaptar o PMDF ao Decus-UUCP

No arquivo UUXQT_DCL.COM:

```
*****
File WRK:[UUCP]UUXQT_DCL.COM_ORIGINAL;2
169 $      IF (f$edit(addressee,"UPCASE") .EQS. "POSTMASTER") THEN -
170          addr = "UUCP_POSTMASTER"
171 $      ON ERROR THEN GOTO FAILED_RMAIL
172 $      save_ver = f$verify(1)
173 $      MAIL/PROTOCOL=UUCP_MAILSHR 'infile' "'addr'"
174 $      save_ver = f$verify(save_ver)
*****
File WRK:[UUCP]UUXQT_DCL.COM_PMDf;3
169 $      ON ERROR THEN GOTO FAILED_RMAIL
170 $      uucp_slave := $pmdf_root:[exe]uucp_slave
171 $      save_ver = f$verify(1)
172 $      define/user sys$input 'infile'
173 $      uucp_slave "'addr'"
174 $      save_ver = f$verify(save_ver)
*****
```

No arquivo UUCP_MASTER.PAS:

```
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;2
11  (*)
12  (*)      Seton R. Droppers (sdroppers@pbs.org)      *)
13  (*)      June 1990                                  *)
14  (*)      Adding support for DECUS UUCP, as          *)
15  (*)      extracted from work by Don Stokes          *)
16  (*)
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;1
11  (*)
*****
```

File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;2

```
38     Uucp_Data_Directory = 'UUCP_SPOOL:';
39     Uucp_Command_Directory = 'UUCP_SPOOL:';
40     Uucp_Command_Prefix = 'C.';
```

File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;1

```
33     Uucp_Data_Directory = 'shell$:[usr.spool.uucp.sys.default.d]';
34     Uucp_Command_Directory = 'shell$:[usr.spool.uucp.sys.default.c]';
35     Uucp_Command_Prefix = 'C.';
```

File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;2

```
298     If I >= 5 then
299         If (Str.Body[I-4] = '.') and
300             ((Str.Body[I-3] = 'U') or (Str.Body[I-3] = 'u')) and
301             ((Str.Body[I-2] = 'U') or (Str.Body[I-2] = 'u')) and
302             ((Str.Body[I-1] = 'C') or (Str.Body[I-1] = 'c')) and
303             ((Str.Body[I] = 'P') or (Str.Body[I] = 'p')) then
304             Str.Length := Str.Length - 5;
305     End; (* Remove_Uucp *)
```

File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;1

```
293     If (Str.Body[I-4] = '.') and
294         ((Str.Body[I-3] = 'U') or (Str.Body[I-3] = 'u')) and
295         ((Str.Body[I-2] = 'U') or (Str.Body[I-2] = 'u')) and
296         ((Str.Body[I-1] = 'C') or (Str.Body[I-1] = 'c')) and
297         ((Str.Body[I] = 'P') or (Str.Body[I] = 'p')) then
298         Str.Length := Str.Length - 5;
299     End; (* Remove_Uucp *)
```

File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;2

```
363     Writeln (Message_File, ' with UUCP/PMDf (DECUS UUCP);');
364     Write (Message_File, '');
```

File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;1

```
357     Writeln (Message_File, ' with UUCP/PMDf (2.0/DECShell/2.4/VMS);');
358     Write (Message_File, '');
```

File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;2

```
392         Uucp_To_Host + '_A' + Extension + ';;';
393     Rem_File_Name.Length := Alfa_Size;
```

File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;1

```
386         Uucp_To_Host + 'a' + Extension + ';;';
387     Rem_File_Name.Length := Alfa_Size;
```

File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;2

```
520     Lib$Sys_Trnlog ('UUCP_HOST_NAME', , Uucp_Host_Name);
521     Lib$Sys_Trnlog ('UUCP_TO_HOST', , Uucp_To_Host);
```

```

*****
File PMDF_ROOT:[SRC.UUCP]UUCP_MASTER.PAS;1
  514   Lib$Sys_Trnlog ('SHELL$UUCP_NAME', , Uucp_Host_Name);
  515   Lib$Sys_Trnlog ('UUCP_TO_HOST', , Uucp_To_Host);
*****

  No arquivo UUCP_SLAVE.PAS:

*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;2
   9   (*      Updated:      Seton Droppers (sdroppers@pbs.org)      *)
  10   (*      June, 1990, for handling DECU UUCP                      *)
  11   (*      Incorporating code from Don Stokes                      *)
  12   (*      (don@gp.govt.nz)                                         *)
  13   (*                                                                *)
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;1
   9   (*                                                                *)
*****
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;2
  233   if Line.Body[1] = '' then Line := Substr(Line, 2, Line.Length-1);
  234   if Line.Body[Line.Length] = '' then Line := Substr(Line, 1, Line.Leng
    th-1);
  235   Line := Line + ' ';
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;1
  229   Line := Line + ' ';
*****
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;2
  248
  249   Procedure Readfromline (var Sender : Vstring);
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;1
  242   Procedure Readfromline (var Sender : Vstring);
*****
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;2
  257   J : Integer;
  258
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;1
  250
*****
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;2
  262   If (Length(Line) > 5) and (Substr(Line, 1, 5) = 'From ') then J:= 6;
  263   If (Length(Line) > 6) and (Substr(Line, 1, 6) = '>From ') then J:= 7;
  264   If (J > 0) then Begin
  265     I := J;
  266     While (Line[I] <> ' ') do I := I + 1;
*****
File PMDF_ROOT:[SRC.UUCP]UUCP_SLAVE.PAS;1

```

```

254      If (Length(Line) > 5) and (Substr(Line,1,5) = 'From ') then Begin
255          I := 6;
256          While (Line[I] <> ' ') do I := I + 1;
*****

```

No arquivo RETURN_UUCP.PAS:

File PMDF_ROOT:[SRC.UUCP]RETURN_UUCP.PAS;2

```

91      Lib$Sys_Trnlog ('UUCP_HOST_NAME', , Uucp_Host_Name);
92      i := vindex (fn, ',');

```

File PMDF_ROOT:[SRC.UUCP]RETURN_UUCP.PAS;1

```

91      Lib$Sys_Trnlog ('SHELL$UUCP_NAME', , Uucp_Host_Name);
92      i := vindex (fn, ',');

```

No arquivo RETURN_UUCP.COM:

File WRK:[PMDf]RETURN_UUCP.COM;2

```

28      /output=3dirlst.tmp uucp_spool:c.*a%%%;*
29      $ !
30      $ directory/before="-6-00"/since="-7-00"/noheader/notrailer/column=1 -
31      /output=6dirlst.tmp uucp_spool:c.*a%%%;*
32      $ !
33      $ directory/before="-9-00"/since="-10-00"/noheader/notrailer/column=1 -
34      /output=9dirlst.tmp uucp_spool:c.*a%%%;*
35      $ !
36      $ directory/before="-12-00"/noheader/notrailer/column=1 -
37      /output=12dirlst.tmp uucp_spool:c.*a%%%;*
38      $ !

```

File WRK:[PMDf]RETURN_UUCP.COM;1

```

28      /output=3dirlst.tmp shell$:[usr.spool.uucp.sys.default.c]c.*a%%%;*/ex
clude=$*$*. *
29      $ !
30      $ directory/before="-6-00"/since="-7-00"/noheader/notrailer/column=1 -
31      /output=6dirlst.tmp shell$:[usr.spool.uucp.sys.default.c]c.*a%%%;*/ex
clude=$*$*. *
32      $ !
33      $ directory/before="-9-00"/since="-10-00"/noheader/notrailer/column=1 -
34      /output=9dirlst.tmp shell$:[usr.spool.uucp.sys.default.c]c.*a%%%;*/ex
clude=$*$*. *
35      $ !
36      $ directory/before="-12-00"/noheader/notrailer/column=1 -
37      /output=12dirlst.tmp shell$:[usr.spool.uucp.sys.default.c]c.*a%%%;*/e
xclude=$*$*. *
38      $ !

```

No arquivo MASTER.COM:

File WRK:[PMDf]MASTER.COM;2

```

305 $ create UUCP_REQUESTS:
306 Wakey wakey!
307 $ goto out1
*****
File WRK:[PMDf]MASTER.COM;1
305 $ uupoll = "$shell$:[usr.lib.uucp]uupoll"
306 $ uupoll 'uucp_to_host'
307 $ goto out1
*****

```

No arquivo UUXQT_DCL.COM:

```

*****
File WRK:[PMDf]UUXQT_DCL.COM;2
169 $      ON ERROR THEN GOTO FAILED_RMAIL
170 $      uucp_slave := $pmdf_root:[exe]uucp_slave
171 $      save_ver = f$verify(1)
172 $      define/user sys$input 'infile'
173 $      uucp_slave "'addr'"
174 $      save_ver = f$verify(save_ver)
*****
File WRK:[PMDf]UUXQT_DCL.COM;1
169 $      IF (f$edit(addressee,"UPCASE") .EQS. "POSTMASTER") THEN -
170 $          addr = "UUCP_POSTMASTER"
171 $      ON ERROR THEN GOTO FAILED_RMAIL
172 $      save_ver = f$verify(1)
173 $      MAIL/PROTOCOL=UUCP_MAILSHR 'infile' "'addr'"
174 $      save_ver = f$verify(save_ver)
*****

```


Apêndice B

Trechos de programas que incorporam a lógica do circuito de arbitramento

Em ulib.c

```
/*
    openline - open the serial port for I/O
*/
openline(name, baud, modem_, parity, xon_xoff)
char *name, *baud, *modem_, *parity, *xon_xoff;
{
    int i, conseguiu;

    sscanf(name, "COM%d", &i);
    select_port(i);
    save_com();
    install_com();
    sscanf(baud, "%d", &i);
    open_com(i, *modem_, *parity, STOPBIT, *xon_xoff);
    /*      alteracao: antes ...dtr_on(); e return(0); */
    conseguiu=dtr_on();
    printf("conseguiu : %d \n", conseguiu);
    return(conseguiu);
} /*openline*/

/*
    closeline - close the serial port down
*/

closeline()
{
    dtr_off();
    close_com();
}
```

```

    restore_com();

} /*closeline*/

Em comm.h:

/*void far dtr_on(); set DTR */
/* Alterei a linha acima para incluir a logica: retornar
ax=0 se conseguiu ou ax=-1 senao */
int far dtr_on();          /* seta dtr , pulsa RTS e le CTS */

Em comm.asm:

;
; DTR_OFF - TURNS OFF DTR TO TELL MODEMS THAT THE TERMINAL HAS GONE AWAY
;          AND TO HANG UP THE PHONE
;
_dtr_off PROC FAR
push bp
mov bp,sp
push si
PUSHF ; SAVE FLAGS
PUSH AX ; SAVE REGS
PUSH DX
PUSH SI
MOV SI,CURRENT_AREA ; SI POINTS TO DATA AREA
TEST INSTALLED[SI],1 ; PORT INSTALLED?
JZ DFX ; ABORT IF NOT
; inicia alteracao o out 08 era o que tinha -> dtr off, rts off, out2 on
    MOV     DX,MCR[SI]
    MOV     AL,0BH ; rts on, dtr on, out2 on -> rts=0
    OUT     DX,AL
    MOV     AL,09  ; rts off, dtr on, out2 on -> rts=1
OUT DX,AL ; ou seja pulsou CK de 0 -> 1
MOV     AL,08  ; rts off, dtr off, out2 on
OUT DX,AL
; fim da alteracao
DFX: POP SI ; RECOVER REGS
POP DX
POP AX
POPF ; RECOVER FLAGS
pop si
mov sp,bp
pop bp
RET
_dtr_off ENDP
;
; DTR_ON - TURNS DTR ON
;
_dtr_on PROC FAR
push bp
mov bp,sp
push si
PUSHF ; SAVE FLAGS

```



```

; * alterei como comentario PUSH AX ; SAVE REGS
PUSH DX
PUSH SI
MOV SI,CURRENT_AREA ; SI POINTS TO DATA AREA
TEST INSTALLED[SI],1 ; PORT INSTALLED?
JZ DNX ; ABORT IF NOT

        MOV     DX,MCR[SI]
        MOV     AL,OBH
        OUT     DX,AL          ; pos RTS em 0
                                ; alteracao inicio
MOV     AL,09H          ; pos RTS em 1 (assim pulsou CK )
        OUT DX,AL
NOP
NOP
NOP          ; espera para estabilizar logica
MOV     DX,MSR[SI]      ; le CTS
IN      AL,DX           ; ou seja CTS = Q
AND     AL,10H
CMP     AL,10H
        JE      CONSEG      ; se igual (CTS=0)conseguiu entao volta ax=0
MOV     AX,-1
JMP     DNX
CONSEG: MOV     AX,0          ; alteracao fim
DNX: POP SI ; RECOVER REGS
POP DX
; * alterei como comentario POP AX
POPF ; RECOVER FLAGS
pop si
mov sp,bp
pop bp
RET ; DONE
_dtr_on      ENDP

```


Apêndice C

Arquivo de Configuração do PMDF para o USPFSC

```
!  
! Hospedeiro Local  
! -----  
! nome do hospedeiro local:  
uspfsc $U@USPFSC.IFQSC.USP.ANSP.BR  
! nome do hospedeiro local com o primeiro subdominio  
uspfsc.ifqsc $U@USPFSC.IFQSC.USP.ANSP.BR  
! nome do hospedeiro local com 1. e 2. subdominios  
uspfsc.ifqsc.usp $U@USPFSC.IFQSC.USP.ANSP.BR  
! nome do hospedeiro local com 1., 2. e 3. subdominios  
uspfsc.ifqsc.usp.ansp $U@USPFSC.IFQSC.USP.ANSP.BR  
! nome do hospedeiro local com todos os dominios  
uspfsc.ifqsc.usp.ansp.br $U@USPFSC.IFQSC.USP.ANSP.BR  
! formas assumidas para USPFSC  
ifqsc $U@USPFSC.IFQSC.USP.ANSP.BR  
ifqsc.usp $U@USPFSC.IFQSC.USP.ANSP.BR  
ifqsc.usp.ansp $U@USPFSC.IFQSC.USP.ANSP.BR  
ifqsc.usp.ansp.br $U@USPFSC.IFQSC.USP.ANSP.BR  
ifqsc.ansp.br $U@USPFSC.IFQSC.USP.ANSP.BR  
! No local para dominio HEPNET  
uspfsc.hepnet $U%USPFSC.HEPNET@DECNET_CHANNEL  
!  
! Rede Local  
! =====  
!  
! formas assumidas para ETHOS ( "usuario@ethos.icmsc.usp.ansp.br" )  
icmsc $U%ETHOS.ICMSC.USP.ANSP.BR@TCP-DAEMON  
icmsc.usp $U%ETHOS.ICMSC.USP.ANSP.BR@TCP-DAEMON  
icmsc.usp.ansp $U%ETHOS.ICMSC.USP.ANSP.BR@TCP-DAEMON  
icmsc.usp.ansp.br $U%ICMSC.USP.ANSP.BR@TCP-DAEMON  
!  
! maquinas VMS  
! -----  
mvaxft.ifqsc.usp.ansp.br $U%MVAXFT.IFQSC.USP.ANSP.BR@DECNET_CHANNEL  
nmr.ifqsc.usp.ansp.br $U%NMR.IFQSC.USP.ANSP.BR@DECNET_CHANNEL
```

```

!
mvaxft.hepnet          $U%MVAXFT.HEPNET@DECNET_CHANNEL
nmr.hepnet             $U%NMR.HEPNET@DECNET_CHANNEL
!
! Sinonimos (nicknames):
!
mvaxft.vms             $U%MVAXFT.IFQSC.USP.ANSP.BR@DECNET_CHANNEL
nmr.vms                $U%NMR.IFQSC.USP.ANSP.BR@DECNET_CHANNEL
mvaxft                 $U%MVAXFT.IFQSC.USP.ANSP.BR@DECNET_CHANNEL
nmr                    $U%NMR.IFQSC.USP.ANSP.BR@DECNET_CHANNEL
!
! maquinas UNIX
! -----
! Fusion
!
floyd.ifqsc.usp.ansp.br $U%floyd.ifqsc.usp.ansp.br@FTCP-DAEMON
emilia.ifqsc.usp.ansp.br $U%emilia.ifqsc.usp.ansp.br@FTCP-DAEMON
dfcmip.ifqsc.usp.ansp.br $U%dfcmip.ifqsc.usp.ansp.br@FTCP-DAEMON
lieip3.ifqsc.usp.ansp.br $U%lieip3.ifqsc.usp.ansp.br@FTCP-DAEMON
neumann.ifqsc.usp.ansp.br $U%neumann.ifqsc.usp.ansp.br@FTCP-DAEMON
meson1.ifqsc.usp.ansp.br $U%meson1.ifqsc.usp.ansp.br@FTCP-DAEMON
meson2.ifqsc.usp.ansp.br $U%meson2.ifqsc.usp.ansp.br@FTCP-DAEMON
meson3.ifqsc.usp.ansp.br $U%meson3.ifqsc.usp.ansp.br@FTCP-DAEMON
ethos.icmsc.usp.ansp.br $U%ethos.icmsc.usp.ansp.br@FTCP-DAEMON
!
! Sinonimos (nicknames):
!
floyd.unix              $U%floyd.ifqsc.usp.ansp.br@FTCP-DAEMON
emilia.unix             $U%emilia.ifqsc.usp.ansp.br@FTCP-DAEMON
ethos.unix              $U%ethos.icmsc.usp.ansp.br@FTCP-DAEMON
dfcmip.unix             $U%dfcmip.ifqsc.usp.ansp.br@FTCP-DAEMON
lieip3.unix             $U%lieip3.ifqsc.usp.ansp.br@FTCP-DAEMON
neumann.unix            $U%neumann.ifqsc.usp.ansp.br@FTCP-DAEMON
meson1.unix             $U%meson1.ifqsc.usp.ansp.br@FTCP-DAEMON
meson2.unix             $U%meson2.ifqsc.usp.ansp.br@FTCP-DAEMON
meson3.unix             $U%meson3.ifqsc.usp.ansp.br@FTCP-DAEMON
!
floyd                   $U%floyd.ifqsc.usp.ansp.br@FTCP-DAEMON
emilia                  $U%emilia.ifqsc.usp.ansp.br@FTCP-DAEMON
ethos                   $U%ethos.icmsc.usp.ansp.br@FTCP-DAEMON
dfcmip                  $U%dfcmip.ifqsc.usp.ansp.br@FTCP-DAEMON
lieip3                  $U%lieip3.ifqsc.usp.ansp.br@FTCP-DAEMON
neumann                 $U%neumann.ifqsc.usp.ansp.br@FTCP-DAEMON
meson1                  $U%meson1.ifqsc.usp.ansp.br@FTCP-DAEMON
meson2                  $U%meson2.ifqsc.usp.ansp.br@FTCP-DAEMON
meson3                  $U%meson3.ifqsc.usp.ansp.br@FTCP-DAEMON
!
! maquinas acessiveis por KA9Q
! -----
xtjan                   $U%uepaesc.embrapa.ansp.br@FTCP-DAEMON
uepaesc                 $U%xtjan.ifqsc.usp.ansp.br@FTCP-DAEMON
!
! Sinonimos (nicknames):

```

```

!
xtjan                                $U%uepaesc.embrapa.ansp.br@FTCP-DAEMON
uepaesc                             $U%xtjan.ifqsc.usp.ansp.br@FTCP-DAEMON
!
! maquinas acessiveis por UUCP
! -----
pcmonic.ifqsc.usp.ansp.br           $U@pcmonic.ifqsc.usp.ansp.br
pcmonic.uucp                        $U@pcmonic.ifqsc.usp.ansp.br
pcmonic                             $U@pcmonic.ifqsc.usp.ansp.br
! maquinas conectadas por PSI-Mail
! -----
!
bruc.bitnet                         $U@BRUC.ANSP.BR
bruc.ansp.br                        $U@BRUC.ANSP.BR
ucif.ansp.br                        $U@UCIF.ANSP.BR
uspfsc.renpac $U@RENPAC.uspfsc
fppsp.renpac $U@RENPAC.fppsp
telebras.renpac $U@RENPAC.telebras
!
! todos os nos locais VMS sao roteados atraves de um canal DECNET
! -----
.vms                                $U%$H.IFQSC.USP.ANSP.BR@DECNET_CHANNEL
!
! todos os nos locais UNIX sao roteados atraves de um canal FTCP (Fusion TCP/IP)
! -----
.unix                               $U%$H.ifqsc.usp.ansp.br@FTCP-DAEMON
!
! todos os nos HEPNET & DECNET sao roteados atraves de um canal DECNet
! -----
.hepnet                             $U%$H.HEPNET@DECNET_CHANNEL
.decnet                             $U%$H.DECNET@DECNET_CHANNEL
!
! todos os nos BITNET sao roteados atraves da FAPESP
! -----
.bitnet                             $U%$H.BITNET@FPSP.FAPESP.ANSP.BR
!
! todos os nos RENPAC sao roteados atraves de um canal PSI
! -----
.renpac                             $U'renpac.$H@PSI_CHANNEL
!
! outros enderecos no dominio IFQSC
! -----
!
.ifqsc $U%$H.IFQSC.USP.ANSP.BR
.ifqsc.usp $U%$H.IFQSC.USP.ANSP.BR
.ifqsc.usp.ansp $U%$H.IFQSC.USP.ANSP.BR
.ifqsc.usp.ansp.br $U%$H.IFQSC.USP.ANSP.BR@DECNET_CHANNEL
!
! outros enderecos no dominio ICMSC
! -----
!
.icmsc $U%$H.ICMSC.USP.ANSP.BR
.icmsc.usp $U%$H.ICMSC.USP.ANSP.BR

```

```

.icmsc.usp.ansp $U/$H.ICMSC.USP.ANSP.BR
.icmsc.usp.ansp.br $U/$H.ICMSC.USP.ANSP.BR@FTCP-DAEMON
!
! todos os outros enderecos sao roteados via FAPESP
! -----
!
fisp.fapesp.ansp.br          $U@FPSP.FAPESP.ANSP.BR
.                            $U@$H@fisp.fapesp.ansp.br

! **** Canais PMDF ****
! -----
! Formato geral: Primeira Linha: tipo/nome do canal e palavras-chave
!                Segunda linha: Nome de hospedeiro oficial alcancavel e
!                sinonimo do hospedeiro local
!                Linhas adicionais: sistemas alcancaveis no canal e seus
!                sinonimos
!                Ultima linha: uma linha em branco
! -----
! Definicao do canal local - * da o nome assumido par o hospedeiro local *
! -----
l 822 bidirectional logging
USPFSC.IFQSC.USP.ANSP.BR

! -----
! Definicoes de canal de correio DECnet
! Nenhuma correspondencia entregue neste canal usara o nome assumido do
! hospedeiro local no From:, pois definimos como nome de hospedeiro
! especifico do canal: USPFSC.HEPNET (resulta: "From: usuario@USPFSC.HEPNET")
! -----
d 733 logging
DECNET_CHANNEL              USPFSC.HEPNET

!
! Definicao do canal PSI_MAIL (Packetnet System Interface MAIL)
! -----
mail_psi logging master_debug slave_debug
PSI_CHANNEL                 USPFSC.IFQSC.USP.ANSP.BR
BRUC.ANSP.BR                RENPAC.11920015
UCIF.ANSP.BR                RENPAC.11920015::FISICA
RENPAC.USPFSC RENPAC.11620020
RENPAC.FPSP RENPAC.11182594
RENPAC.BRUC RENPAC.11920015
RENPAC.TELEBRAS RENPAC.11920020

!
! canal DECnet Phonenet (canal de conexao com FPSP)
! -----
dn_fpisp master network immediate logging master_debug slave_debug
fisp.fapesp.ansp.br        ifqsc.usp.ansp.br

!
! canal Fusion tcp/ip (canal de conexao com sistemas UNIX)

```

```
! -----  
ftcp_local single_sys smtp master_debug slave_debug  
FTCP-DAEMON  
  
! Novo canal para decus UUCP com alteracoes para o PMDF 3.1  
! -----  
! para chegada  
uucp_gateway uucp slave master_debug slave_debug  
uucp_gateway  
  
! para saida  
uucp_pcmonic uucp master slave_debug master_debug  
uucp_pcmonic  
pcmonic.ifqsc.usp.ansp.br pcmonic
```


Apêndice D

Arquivo de configuração do Sendmail para a estação de trabalho floyd

```
#####
#
# Arquivo de configuracao do Sendmail para
# "MAQUINAS PRINCIPAIS"
# Deve-se instalar este arquivo como /etc/sendmail.cf
# se a sua maquina e' o principal (ou unico)
# retransmissor de correspondencia do seu dominio.
# Entao edite o arquivo para adaptar a configuracao
# de sua rede.
#
# @(#)main.mc 1.17 90/01/04 SMI
#

### informacao local

# meu nome de hospedeiro oficial
# Voce tem duas opcoes aqui. Se voce quer que a maquina "entron-
# camento" identifique-se como dominio use:
#Dj$m
# Se voce quer que a maquina "entroncamento" apareca dentro do dominio, use:
Dj$w.$m
# Se possuir um nome de hospedeiro completamente qualificado, use:
#Dj$w

# principal despachador-retransmissor (mailer-relay)
DMrelay

# principal hospedeiro retransmissor: usa o $M despachador para enviar
# correspondencia para outros dominios
DRuspfsc
CRuspfsc

# Se voce quer carregar os hospedeiros que permitem correspondencia, use:
```

```

# FS /usr/lib/mailhosts
# e entao troque as ocorrencias de $%y para $=S.
# Senao, o mapeamento hosts.byname do NIS e' assumido se o NIS
# estiver rodando (senao estiver, o arquivo/etc/hosts).

# opcoes provavelmente desejadas no hospedeiro de correspondencia

# ponto de checagem da fila apos esse nro de recipientes(destinatarios)
OC10

# recusa o envio de mensagens vazias para mais do que esse nro de recipientes
Ob10

#####
#
# Informacao de Configuracao Geral

# nomes de dominio local
#
# Se o dominio do NIS e' diferente do nome que voce deseja que apareca
# nos cabecalhos de correspondencia, edite as linhas abaixo para refle-
# tir os dominios que voce deseja.
# Note que o primeiro componente do nome de dominio do NIS e' retirado
# a menos que o nome de dominio comece com "." ou "+".
# Exemplo:
# DmCS.Podunk.EDU
# Cm cs cs.Podunk.EDU
#
# O Dm e' o valor estabelecido para as correspondencias que saem.
# o Cm e' o valor que e' aceito para as correspondencias que chegam.
# Normalmente sao os mesmos, porem voce pode querer possuir mais de
# uma linha Cm para reconhecer mais do que um nome de dominio durante
# uma transicao.

# Numero da versao do arquivo de configuracao
DVSMI-4.1

###  Macros Padroes

# nome usado para enviar mensagens com erro
DnMailer-Daemon
# formato de cabecalho do UNIX
DlFrom $g $d
# caracteres delimitadores (operadores)
Do.:%@!^=/[ ]
# formato de um nome completo
Dq$g$x ($x)$.
# Mensagem usada para o "login" do SMTP
De$j Sendmail $v/$V ready at $b

###  Opcoes

```

```

# Modo Remoto - envia pelo servidor se o diretorio caixa-postal esta' montado
OR
# localizacao do arquivo de apelidos
OA/etc/aliases
# modo de entrega assumido (entrega em "background")
Odbackground
# reconstroe o arquivo de apelidos automaticamente
OD
# modo de protecao dos arquivos temporarios -- 0600 para seguranca, 0644 para permissao
OF0600
# GID (identificacao de grupo) assumido
Og1
# localizacao do arquivo de ajuda
OH/usr/lib/sendmail.hf
# nivel de registro
OL15
# mensagens assumidas - estilo antigo
Oo
# "Postmaster" local - gerente de correspondencia local (p/ replicas de erro)
OPPostmaster
# diretorio da fila de correspondencia
OQ/usr/spool/mqueue
# tempo de expiracao para leitura no protocolo SMTP
Or15m
# arquivo de estado - nenhum
OS/etc/sendmail.st
# por seguranca enfileira tudo antes de iniciar a transmissao
Os
# apos ficarem enfileiradas esse tempo, as mensagens retornam ao
# que enviou
OT3d
# UID (identificacao de usuario) assumido
Oul

### Precedencia de Mensagens
Pfirst-class=0
Pspecial-delivery=100
Pjunk=-100

### Usuarios de Confianca
T root daemon uucp

### Formato dos cabecalhos
H?P?Return-Path: <$g>
HReceived: $?sfrom $s $.by $j ($v/$V)
id $i; $b
H?D?Resent-Date: $a
H?D?Date: $a
H?F?Resent-From: $q
H?F?From: $q
H?x?Full-Name: $x
HSubject:
H?M?Resent-Message-Id: <$t.$i@ $j>

```

H?M?Message-Id: <\$t.\$i@\$\$j>

HErrors-To:

#####

Regras de Tradução

#####

Pre-escrita do campo Remetente

S1

Nada e' necessario.

Pre-escrita do campo Destinatario

S2

Nada e' necessario.

Colocacao dos nomes na forma Canonica

O formato interno dos nome dentro das regras de reescrita e':

qualquer_coisa<@hospedeiro.dominio.dominio...>qualquer_coisa

Coloca-se todo tipo de nome nesse formato, exceto nomes

locais, que nao possuem parte de hospedeiro. A razao para "<>"

e' que todo nome de hospedeiro relevante pode estar em frente

do nome (para encaminhamento fonte), ou atras (forma normal).

Coloca-se o que e' para encaminhar dentro de "<>" para ser

facil de achar.

#

S3

tratando o caso especial "from:<>"

R\$*<>\$* \$\$Q transforma no "token" magico

canonicalizacao de texto basica

R\$*<\$+>\$* \$2 analise RFC 822 basica

para certificar que a sintaxe <@a,@b,@c:user@d>

e' facil de analisar - desfaz depois

RQ\$+,,\$+:\$+ @1:\$2:\$3 troca toda ",", por ":"

RQ\$+:\$+ \$\$Q>6<@1>:\$2 encaminhamento fonte canonico

R\$+:\$*;@+ \$\$Q1:\$2;@3 sintaxe de lista

R\$+@+ \$:\$1<@2> enfase no dominio

R\$+<\$+@+> \$1\$2<@3> move o enfoque

R\$+<@+> \$\$Q>6\$1<@2> ja canonico

conversao de nomes estilo antigo para nomes estilo dominio

Todos os nomes estilo antigo sao analisados da esquerda para

direita, sem precedencia.

R\$-!\$+ \$\$Q>6\$2<@1.uucp> hospedeiro_uucp!usuario

R\$-.\$+!\$+ \$\$Q>6\$3<@1.\$2> hospedeiro.dominio!usuario

R\$+/%\$+ \$\$Q>3\$1@2 usuario/hospedeiro

```
# Reescrita final para sair
S4
R$*<@$.uucp> $2!$1 u@h.uucp => h!u
R$+ $: $>9 $1 limpa denominacao
R$*<$+>$* $1$2$3 desfaz "<>"
```

```
# Limpa o nome para passar para algum despachador
# (mas deixa com "<>")
S9
R$=w!@ $@$w!$n
R@ $@$n trata do erro na denominacao: <>
R$*<$*LOCAL>$* $1<$2$m>$3 troca informacao local
R<@$+>$*:$+:$+ <@$1>$2,$3:$4 <encaminhamento> canonico
```

```
#####
# Regras de Reescrita
```

```
# conversoes especiais locais
S6
R$*<@$*$=m>$* $1<@$2LOCAL>$4 converte para o dominio LOCAL
# para as com dominio ifqsc.usp.ansp.br que nao estejam no arquivo hosts
# ($/y) serem mandadas para o relay-host (USPFSC)
R$*<@$%y.$=m>$* $1<@$2.LOCAL>$4 converte para dominio LOCAL
```

```
# Especificacao dos despachadores local e de programa
```

```
Mlocal, P=/bin/mail, F=rlsDFMmnP, S=10, R=20, A=mail -d $u
Mprog, P=/bin/sh, F=lsDFMeuP, S=10, R=20, A=sh -c $u
```

```
S10
# Nada e' necessario
```

```
S20
# Nada e' necessario
```

```
#####
####
#### Especificacao do Despachador Ethernet
####
#### Mensagens processadas por essa configuracao estao assumidas
#### permanecerem no mesmo dominio. Realmente nao ha nada a fazer
#### particularmente com Ethernet - o nome e' historico
```

```
Mether, P=[TCP], F=msDFMuCX, S=11, R=21, A=TCP $h
```

```
S11
R$*<@$+>$* $@$1<@$2>$3 ja correto
R$+ $@$1<@$w> acrescenta nosso nome de hospedeiro
```

```
S21
# Nada e' necessario
```

```
#####
```

####

Especificacao do Despachador relay

####

Mrelay, P=[TCP], F=msDFMuCX, S=31, R=41, A=TCP \$h

S31

R\$*<@\$.LOCAL>\$* \$0\$1<0\$2.\$m>\$3 ja correto

R\$*<@+>\$* \$0\$1<0\$2>\$3 ja correto

R\$+ \$0\$1<0\$w> acrescenta nosso nome de hospedeiro

S41

Nada e' necessario

#####

#

Conjunto 0 de regras

#

Este e' o conjunto de regras que determina o despachador

para o qual a denominacao deve ser passada

O conjunto de regras 30 chama o conjunto de regras 3 e entao o 0.

S30

R\$* \$: \$>3 \$1 primeira canonicalizacao

R\$* \$0 \$>0 \$1 e entao roda o conjunto de regras 0

S0

Na entrada o nome ja foi posto na forma canonica pelo conjunto de regras 3

tratando casos especiais

R0 \$#local \$:\$n tratando da forma <>

coloca o nome de hospedeiro local para LOCAL

R\$*<\$*=w.LOCAL>\$* \$1<\$2LOCAL>\$4 este_hospedeiro.LOCAL

R\$*<\$*=w.uucp>\$* \$1<\$2LOCAL>\$4 este_hospedeiro.uucp

R\$*<\$*=w>\$* \$1<\$2LOCAL>\$4 este_hospedeiro

Correspondencia destinada ao entroncamento de dominio (US)

R\$*<0LOCAL> \$0\$>30\$1 retira noss nome e tenta de novo

R<0LOCAL>:\$+ \$0\$>30\$1 tenta de novo apos retirar encaminhamento

Para especificacao numerica

R<0[\$+]>:\$* \$:\$>9 <0[\$1]>:\$2 limpa e entao continua...

R<0[\$+]>:\$* \$#ether \$0[\$1] \$:\$2 especificacao numerica internet

R<0[\$+]>:\$* \$#ether \$0[\$1] \$:\$2 especificacao numerica internet

R\$*<0[\$+]> \$#ether \$0[\$2] \$:\$1 especificacao numerica internet

entrega a hospedeiros ethernet conhecidos explicitamente especificados no

nosso dominio

R\$*<0\$%y.LOCAL>\$* \$#ether \$0\$2 \$:\$1<0\$2>\$3 usuario@hospedeiro.sun.com

hospedeiro_ether.uucp e' tratado como hospedeiro_ether.\$m aqui.

Isso permite denominar de um local uucp foo!sun!hospedeiro_ether!usuario

R\$*<0\$%y.uucp>\$* \$#ether \$0\$2 \$:\$1<0\$2>\$3 usuario@hospedeiro_ether.uucp

```

# Explicitamente especifica nomes no nosso dominio --
# os quais os nomes de hospedeiros nao sao conhecidos
R$*<@$.LOCAL>$* $#error $:Never heard of host $2 in domain $m

# Limpa as denominacoes para uso externo -- retira LOCAL, encaminhamento ,=>:
R$* $:>9 $1 entao continua...

# regra especifica para os dominios com que tem ligacao direta
# nao precisa passar para o relay-host

# a maquina de entroncamento de correio que possui como sinonimo o nome
# do dominio que faz parte, e' com quem e' feita a secao ether smtp
# para usuario@icmsc.usp.ansp.br
R$*<@icmsc.usp.ansp.br>$* $#ether $@icmsc $:$1<@icmsc.usp.ansp.br>$2
R$*<@$.icmsc.usp.ansp.br>$* $#ether $@icmsc $:$1<@ $2.icmsc.usp.ansp.br>$3

# para usuario@uepaesc.embrapa.ansp.br
R$*<@uepaesc.embrapa.ansp.br>$* $#ether
$@uepaesc $:$1<@uepaesc.embrapa.ansp.br>$2
R$*<@$.uepaesc.embrapa.ansp.br>$* $#ether
$@uepaesc $:$1<@ $2.uepaesc.embrapa.ansp.br>$3

# Passa os nomes validos ao hospedeiro retransmissor
R$*<@$.+>$* $#M $@R $:$1<@ $2.$3>$4 usuario@qualquer.dominio

# Todas as denominacoes nas regras acima sao absolutas
# (qualificada completamente com dominios)
# As denominacoes abaixo podem ser parcialmente qualificadas
# entrega a hospedeiros ethernet conhecidos
R$*<@$/y>$* $#ether $@ $2 $:$1<@ $2>$3 usuario@hospedeiro_ether

# outros nomes nao locais que nao tem para onde ir; voltam ao remetente
R$*<@$.+>$* $#error $:Unknown domain $3
# R$*<@$.+>$* $#error $:Never heard of $2 in domain $m
# para enviar a usuario@mvaxft atraves do USPFSC
R$*<@$.+>$* $#M $@R $:$1<@ $@>$#

R$*@ $* $#error $:I don't understand $1@ $2

# Denominacoes locais com % nao sao realmente locais
R$+ %$+ $@ $>30 $1@ $2 troca "%" por "@", e tenta de novo

# tudo o que sobra e' nome local
R$+ $#local $:$1 nomes locais

```


Apêndice E

Arquivo de configuração do Sendmail para a estação de trabalho emilia

```
#####  
#  
# Arquivo de configuracao do Sendmail para  
# "MAQUINAS SUBSIDIARIAS"  
# Deve-se instalar este arquivo como /etc/sendmail.cf  
# se a sua maquina e' o principal (ou unico)  
# retransmissor de correspondencia do seu dominio.  
# Entao edite o arquivo para adaptar a configuracao  
# de sua rede.  
#  
# @(#)main.mc 1.17 90/01/04 SMI  
#  
  
### informacao local  
  
# meu nome de hospedeiro oficial  
# Voce tem duas opcoes aqui. Se voce quer que a maquina "entron-  
# camento" identifique-se como dominio use:  
#Dj$m  
# Se voce quer que a maquina "entroncamento" apareca dentro do dominio, use:  
Dj$w.$m  
# Se possuir um nome de hospedeiro completamente qualificado, use:  
#Dj$w  
  
# principal despachador-retransmissor (mailer-relay) - a escolha tipica  
# e' "ddn" se voce esta na Defense Data Network (e.g. Arpanet or Milnet)  
DMsmartuucp  
  
# principal hospedeiro retransmissor: usa o $M despachador para enviar  
# correspondencia para outros dominios  
DRddn-gateway  
CRddn-gateway
```

```

# Se voce quer carregar os hospedeiros que permitem correspondencia, use:
# FS /usr/lib/mailhosts
# e entao troque as ocorrencias de $%y para $=S.
# Senao, o mapeamento hosts.byname do NIS e' assumido se o NIS
# estiver rodando (senao estiver, o arquivo/etc/hosts).

# opcoes provavelmente desejadas no hospedeiro de correspondencia

# ponto de checagem da fila apos esse nro de recipientes(destinatarios)
OC10

# recusa o envio de mensagens vazias para mais do que esse nro de recipientes
Ob10

#####
#
# Informacao de Configuracao Geral

# nomes de dominio local
#
# Se o dominio do NIS e' diferente do nome que voce deseja que apareca
# nos cabecalhos de correspondencia, edite as linhas abaixo para refle-
# tir os dominios que voce deseja.
# Note que o primeiro componente do nome de dominio do NIS e' retirado
# a menos que o nome de dominio comece com "." ou "+".
# Exemplo:
# DmCS.Podunk.EDU
# Cm cs cs.Podunk.EDU
#
# O Dm e' o valor estabelecido para as correspondencias que saem.
# o Cm e' o valor que e' aceito para as correspondencias que chegam.
# Normalmente sao os mesmos, porem voce pode querer possuir mais de
# uma linha Cm para reconhecer mais do que um nome de dominio durante
# uma transicao.

# Numero da versao do arquivo de configuracao
DVSMI-4.1

###  Macros Padroes

# nome usado para enviar mensagens com erro
DnMailer-Daemon
# formato de cabecalho do UNIX
DlFrom $g $d
# caracteres delimitadores (operadores)
Do.:%@!^=/[ ]
# formato de um nome completo
Dq$g$x ($x)$
# Mensagem usada para o "login" do SMTP
De$j Sendmail $v/$V ready at $b

###  Opcoes

```

```

# Modo Remoto - envia pelo servidor se o directorio caixa-postal esta' montado
OR
# localizacao do arquivo de apelidos
OA/etc/aliases
# modo de entrega assumido (entrega em "background")
Odbackground
# reconstroe o arquivo de apelidos automaticamente
OD
# modo de protecao dos arquivos temporarios -- 0600 para seguranca, 0644 para permissao
OF0600
# GID (identificacao de grupo) assumido
Og1
# localizacao do arquivo de ajuda
OH/usr/lib/sendmail.hf
# nivel de registro
OL15
# mensagens assumidas - estilo antigo
Oo
# "Postmaster" local - gerente de correspondencia local (p/ replicas de erro)
OPPostmaster
# directorio da fila de correspondencia
OQ/usr/spool/mqueue
# tempo de expiracao para leitura no protocolo SMTP
Or15m
# arquivo de estado - nenhum
OS/etc/sendmail.st
# por seguranca enfileira tudo antes de iniciar a transmissao
Os
# apos ficarem enfileiradas esse tempo, as mensagens retornam ao
# que enviou
OT3d
# UID (identificacao de usuario) assumido
Oui

### Precedencia de Mensagens
Pfirst-class=0
Pspecial-delivery=100
Pjunk=-100

### Usuarios de Confianca
T root daemon uucp

### Formato dos cabecalhos
H?P?Return-Path: <$g>
HReceived: $?sfrom $s $.by $j ($v/$V)
id $i; $b
H?D?Resent-Date: $a
H?D?Date: $a
H?F?Resent-From: $q
H?F?From: $q
H?x?Full-Name: $x
HSubject:

```

```
H?M?Resent-Message-Id: <$t.$i@$j>
H?M?Message-Id: <$t.$i@$j>
HErrors-To:
```

```
#####
###  Regras de Tradução ###
#####
```

```
# Pre-escrita do campo Remetente
S1
# Nada e' necessario.
```

```
# Pre-escrita do campo Destinatario
S2
# Nada e' necessario.
```

```
# Colocacao dos nomes na forma Canonica
```

```
# O formato interno dos nome dentro das regras de reescrita e':
#   qualquer_coisa<@hospedeiro.dominio.dominio...>qualquer_coisa
# Coloca-se todo tipo de nome nesse formato, exceto para nomes
# locais, que nao possuem parte de hospedeiro. A razao para "<>"
# e' que todo nome de hospedeiro relevante pode estar em frente
# do nome (para encaminhamento fonte), ou atras (forma normal).
# Coloca-se o que e' para encaminhar dentro de "<>" para ser
# facil de achar.
#
S3
```

```
# tratando o caso especial "from:<>"
R$*<>$* $$$ transforma no "token" magico
```

```
# canonicalizacao de texto basica
R$*<$+>$* $2 analise RFC 822 basica
```

```
# para certificar que a sintaxe <@a,@b,@c:user@d>
# e' facil de analisar - desfaz depois
RQ$+,$+:$+ Q$1:$2:$3 troca toda "," por ":"
RQ$+:$+ Q$>6<Q$1>:$2 encaminhamento fonte canonico
```

```
R$+:$+;Q$+ Q$1:$2;Q$3 sintaxe de lista
R$+Q$+ $:$1<Q$2> enfase no dominio
R$+<$+Q$+> $1$2<Q$3> move o enfoque
R$+<Q$+> Q$>6$1<Q$2> ja canonico
```

```
# conversao de nomes estilo antigo para nomes estilo dominio
# Todos os nomes estilo antigo sao analisados da esquerda para
# direita, sem precedencia.
R$-!$+ Q$>6$2<Q$1.uucp> hospedeiro_uucp!usuario
R$-,$+!$+ Q$>6$3<Q$1.$2> hospedeiro.dominio!usuario
R$+%$+ Q$>3$1Q$2 usuario%hospedeiro
```

```
# Reescrita final para sair
S4
R$+<@$.uucp> $2!$1 u@h.uucp => h!u
R$+ $: $>9 $1 limpa denominacao
R$*<$+>$* $1$2$3 desfaz "<>"
```

```
# Limpa o nome para passar para algum despachador
# (mas deixa com "<>")
S9
R$=w!@ $Q$w!$n
R@ $Q$n trata do erro na denominacao: <>
R$*<$*LOCAL>$* $1<$2$m>$3 troca informacao local
R<@ $+>$*.$+:$+ <@ $1>$2,$3:$4 <encaminhamento> canonico
```

```
#####
# Regras de Reescrita
```

```
# conversoes especiais locais
S6
R$*<@ $* $=m>$* $1<@ $2LOCAL>$4 converte para o dominio LOCAL
```

```
# Especificacao dos despachadores local e de programa
```

```
Mlocal, P=/bin/mail, F=rlsDFMmnP, S=10, R=20, A=mail -d $u
Mprog, P=/bin/sh, F=lsDFMeuP, S=10, R=20, A=sh -c $u
```

```
S10
# Nada e' necessario
```

```
S20
# Nada e' necessario
```

```
#####
#####
##### Especificacao do Despachador Ethernet
#####
##### Mensagens processadas por essa configuracao estao assumidas
##### permanecerem no mesmo dominio. Realmente nao ha nada a fazer
##### particularmente com Ethernet - o nome e' historico
```

```
Mether, P=[TCP], F=msDFMuCX, S=11, R=21, A=TCP $h
S11
R$*<@ $+>$* $Q$1<@ $2>$3 ja correto
R$+ $Q$1<@ $w> acrescenta nosso nome de hospedeiro
```

```
S21
# Nada e' necessario
```

```
#####
#
```

```

# Conjunto 0 de regras
#
# Este e' o conjunto de regras que determina o despachador
# para o qual a denominacao deve ser passada

# O conjunto de regras 30 chama o conjunto de regras 3 e entao o 0.
S30
R$* $: $>3 $1 primeira canonicalizacao
R$* $0 $>0 $1 e entao roda o conjunto de regras 0

S0
# Na entrada o nome ja foi posto na forma canonica pelo conjunto de regras 3
# tratando casos especiais
R0 $#local $:$n tratando da forma <>

# apaga informacao local redundante
R$*<$*=w.LOCAL>$* $1<$2>$4 este_hospedeiro.LOCAL
R$*<@LOCAL>$* $1<@m>$2 hospedeiro = entroncamento do dominio
R$*<$*=w.uucp>$* $1<$2>$4 este_hospedeiro.uucp
R$*<$*=w>$* $1<$2>$4 este_hospedeiro

# Para especificacao numerica
R<@[$+]>:$* $:$>9 <@[$1]>:$2 limpa e entao continua...
R<@[$+]>:$* $#ether @[$1] $:$2 especificacao numerica internet
R<@[$+]>,$* $#ether @[$1] $:$2 especificacao numerica internet
R$*<@[$+]> $#ether @[$2] $:$1 especificacao numerica internet

R$*<$*.$>$* $1<$2>$3 descarta ponto inicial
R<@>:$* @0$>30$1 tenta novamente apos tirar roteamento
R$*<@> @0$>30$1 tira lixo nulo e tenta novamente

#####
### Parte dependente de maquina do conjunto 0 de regras
###

# otimiza nomes de hospedeiros ethernet conhecidos
R$*<@%y.LOCAL>$* $#ether @0$2 $:$1<@0$2>$3 usuario@hospedeiro.aqui

# outros nomes nao locais que nao tem para onde ir; voltam ao remetente
R$*<@$.>$* $#error $:Unknown domain $3
R$*<@+>$* $#error $:Never heard of $2 in domain $m
R$*@0$* $#error $:I don't understand $1@0$2

# Denominacoes locais com % nao sao realmente locais
R$+/%$+ @0$>30$1@0$2 troca "%" por "@", e tenta de novo

# tudo o que sobra e' nome local
R$+ $#local $:$1 nomes locais

```

Apêndice F

Arquivo de configuração do Sendmail para a estação de trabalho ethos

```
#####
#
# Arquivo de configuracao do Sendmail para
#           "MAQUINAS PRINCIPAIS"
#       Deve-se instalar este arquivo como /etc/sendmail.cf
#       se a sua maquina e' o principal (ou unico)
#       retransmissor de correspondencia do seu dominio.
#       Entao edite o arquivo para adaptar a configuracao
#       de sua rede.
#
# @(#)main.mc 1.17 90/01/04 SMI
#

### informacao local

# meu nome de hospedeiro oficial
# Voce tem duas opcoes aqui. Se voce quer que a maquina "entron-
# camento" identifique-se como dominio use:
#Dj$m
# Se voce quer que a maquina "entroncamento" apareca dentro do dominio, use:
#Dj$w.$m
# Se possuir um nome de hospedeiro completamente qualificado, use:
#Dj$w

# principal despachador-retransmissor (mailer-relay)
DMrelay

# principal hospedeiro retransmissor: usa o $M despachador para enviar
# correspondencia para outros dominios
DRfloyd
CRfloyd

# Se voce quer carregar os hoppedeiros que permitem correspondencia, use:
```

```

# FS /usr/lib/mailhosts
# e entao troque as ocorrencias de $%y para $=S.
# Senao, o mapeamento hosts.byname do NIS e' assumido se o NIS
# estiver rodando (senao estiver, o arquivo/etc/hosts).

# opcoes provavelmente desejadas no hospedeiro de correspondencia

# ponto de checagem da fila apos esse nro de recipientes(destinatarios)
OC10

# recusa o envio de mensagens vazias para mais do que esse nro de recipientes
Ob10

#####
#
# Informacao de Configuracao Geral

# nomes de dominio local
#
# Se o dominio do NIS e' diferente do nome que voce deseja que apareca
# nos cabecalhos de correspondencia, edite as linhas abaixo para refle-
# tir os dominios que voce deseja.
# Note que o primeiro componente do nome de dominio do NIS e' retirado
# a menos que o nome de dominio comece com um "." ou "+".
# Exemplo:
# DmCS.Podunk.EDU
# Cm cs cs.Podunk.EDU
#
# O Dm e' o valor estabelecido para as correspondencias que saem.
# o Cm e' o valor que e' aceito para as correspondencias que chegam.
# Normalmente sao os mesmos, porem voce pode querer possuir mais de
# uma linha Cm para reconhecer mais do que um nome de dominio durante
# uma transicao.
Dmicmsc.usp.ansp.br
Cmicmsc icmsc.usp.ansp.br

# Numero da versao do arquivo de configuracao
DVSMI-4.1

###  Macros Padroes

# nome usado para enviar mensagens com erro
DnMailer-Daemon
# formato de cabecalho do UNIX
DlFrom $g $d
# caracteres delimitadores (operadores)
Do.:%@!~=/[ ]
# formato de um nome completo
Dq$g$x ($x)$.
# Mensagem usada para o "login" do SMTP
De$j Sendmail $v/$V ready at $b

```


Opcoes

```
# Modo Remoto - envia pelo servidor se o diretorio caixa-postal esta' montado
OR
# localizacao do arquivo de apelidos
OA/etc/aliases
# modo de entrega assumido (entrega em "background")
Odbackground
# reconstroe o arquivo de apelidos automaticamente
OD
# modo de protecao dos arquivos temporarios -- 0600 para seguranca, 0644 para permissao
OF0600
# GID (identificacao de grupo) assumido
Og1
# localizacao do arquivo de ajuda
OH/usr/lib/sendmail.hf
# nivel de registro
OL15
# mensagens assumidas - estilo antigo
Oo
# "Postmaster" local - gerente de correspondencia local (p/ replicas de erro)
OPPostmaster
# diretorio da fila de correspondencia
OQ/usr/spool/mqueue
# tempo de expiracao para leitura no protocolo SMTP
Or15m
# arquivo de estado - nenhum
OS/etc/sendmail.st
# por seguranca enfileira tudo antes de iniciar a transmissao
Os
# apos ficarem enfileiradas esse tempo, as mensagens retornam ao
# que enviou
OT3d
# UID (identificacao de usuario) assumido
Oui
```

Precedencia de Mensagens

```
Pfirst-class=0
Pspecial-delivery=100
Pjunk=-100
```

Usuarios de Confianca

```
T root daemon uucp
```

Formato dos cabecalhos

```
H?P?Return-Path: <$g>
HReceived: $?sfrom $s $.by $j ($v/$V)
id $i; $b
H?D?Resent-Date: $a
H?D?Date: $a
H?F?Resent-From: $q
H?F?From: $q
H?x?Full-Name: $x
```

```

HSubject:
H7M?Resent-Message-Id: <$t.$i0$j>
H7M?Message-Id: <$t.$i0$j>
HErrors-To:

#####
###  Regras de Traducaao ###
#####

# Pre-escrita do campo Remetente
S1
# Nada e' necessario.

# Pre-escrita do campo Destinatario
S2
# Nada e' necessario.

# Colocacao dos nomes na forma Canonica

# O formato interno dos nome dentro das regras de reescrita e':
#   qualquer_coisa<@hospedeiro.dominio.dominio...>qualquer_coisa
# Coloca-se todo tipo de nome nesse formato, exceto para nomes
# locais, que nao possuem parte de hospedeiro. A razao para "<>"
# e' que todo nome de hospedeiro relevante pode estar em frente
# do nome (para encaminhamento fonte), ou atras (forma normal).
# Coloca-se o que e' para encaminhar dentro de "<>" para ser
# facil de achar.
#
S3

# tratando o caso especial "from:<>"
R$*<>$* $00  transforma no "token" magico

# canonicalizacao de texto basica
R$*<$+>$* $2 analise RFC 822 basica

# para certificar que a sintaxe <@a,@b,@c:user@d>
# e' facil de analisar - desfaz depois
R0$+,$+:$+ 0$1:$2:$3 troca toda "," por ":"
R0$+:$+ 0$0$>6<0$1>:$2 encaminhamento fonte canonico

R$+:$+;0$+ 0$0$1:$2;0$3 sintaxe de lista
R$+0$+ $:$1<0$2> enfase no dominio
R$+<$+0$+> $1$2<0$3> move o enfoque
R$+<0$+> 0$0$>6$1<0$2> ja canonico

# conversao de nomes estilo antigo para nomes estilo dominio
# Todos os nomes estilo antigo sao analisados da esquerda para
# direita, sem precedencia.
R$-!$+ 0$0$>6$2<0$1.uucp> hospedeiro_uucp!usuario
R$-.$+!$+ 0$0$>6$3<0$1.$2> hospedeiro.dominio!usuario

```

R\$+/%\$+ \$0\$>3\$10\$2 usuario%/hospedeiro

Reescrita final para sair

S4

R\$+<0\$+.uucp> \$2!\$1 u@h.uucp => h!u

R\$+ \$: \$>9 \$1 limpa denominacao

R\$*<\$+>\$* \$1\$2\$3 desfaz "<>"

Limpa o nome para passar para algum despachador

(mas deixa com "<>"

S9

R\$=w!0 \$0\$w!\$n

R0 \$0\$n trata do erro na denominacao: <>

R\$*<\$*LOCAL>\$* \$1<\$2\$m>\$3 troca informacao local

R<0\$+>\$*.\$+:.\$+ <0\$1>\$2,\$3:\$4 <encaminhamento> canonico

#####

Regras de Reescrita

conversoes especiais locais

S6

R\$*<0\$*\$=m>\$* \$1<0\$2LOCAL>\$4 converte para o dominio LOCAL

Especificacao dos despachadores local e de programa

Mlocal, P=/bin/mail, F=rlsDFMmnP, S=10, R=20, A=mail -d \$u

Mprog, P=/bin/sh, F=lsDFMeuP, S=10, R=20, A=sh -c \$u

S10

Nada e' necessario

S20

Nada e' necessario

#####

#####

Especificacao do Despachador Ethernet

#####

Mensagens processadas por essa configuracao estao assumidas

permanecerem no mesmo dominio. Realmente nao ha nada a fazer

particularmente com Ethernet - o nome e' historico

Mether, P=[TCP], F=msDFMuCX, S=11, R=21, A=TCP \$h

S11

R\$*<0\$+>\$* \$0\$1<0\$2>\$3 ja correto

R\$+ \$0\$1<0\$w> acrescenta nosso nome de hospedeiro

S21

Nada e' necessario

```
#####
```

```
####
```

```
##### Especificacao do Despachador relay
```

```
####
```

```
Mrelay, P=[TCP], F=msDFMuCX, S=31, R=41, A=TCP $h
```

```
S31
```

```
# Quando quizer que os froms das maquinas que pertencem a um mesmo
# dominio saiam apenas com o nome do dominio e nao da maquina. Isto e'
# util para alias para usuarios de um mesmo dominio sem NIS terem o nome
# da maquina retirado do from. E quando chegar para a maquina que atende
# ao dominio exista um alias para cada usuario especificando em que
# maquina ele quer receber o mail
```

```
R$*<@$.LOCAL>$* $Q$1<Q$m>$3 retira maquina poe dominio
```

```
R$*<Q$+>$* $Q$1<Q$2>$3 ja correto
```

```
R$+ $Q$1<Q$w> acrescenta nosso nome de hospedeiro
```

```
S41
```

```
# Nada e' necessario
```

```
#####
```

```
#
```

```
# Conjunto 0 de regras
```

```
#
```

```
# Este e' o conjunto de regras que determina o despachador
# para o qual a denominacao deve ser passada
```

```
# O conjunto de regras 30 chama o conjunto de regras 3 e entao o 0.
```

```
S30
```

```
R$* $: $>3 $1 primeira canonicalizacao
```

```
R$* $Q $>0 $1 e entao roda o conjunto de regras 0
```

```
S0
```

```
# Na entrada o nome ja foi posto na forma canonica pelo conjunto de regras 3
# tratando casos especiais
```

```
RQ $#local $:$n tratando da forma <>
```

```
# coloca o nome de hospedeiro local para LOCAL
```

```
R$*<$=$w.LOCAL>$* $1<$2LOCAL>$4 este_hospedeiro.LOCAL
```

```
R$*<$=$w.uucp>$* $1<$2LOCAL>$4 este_hospedeiro.uucp
```

```
R$*<$=$w>$* $1<$2LOCAL>$4 este_hospedeiro
```

```
# Correspondencia destinada ao entroncamento de dominio
```

```
R$*<@LOCAL> $Q$>30$1 retira noss nome e tenta de novo
```

```
R<@LOCAL>:$+ $Q$>30$1 tenta de novo apos retirar encaminhamento
```

```
# Para especificacao numerica
```

```
R<Q[$+]>:$* $:$>9 <Q[$1]>:$2 limpa e entao continua...
```

```
R<Q[$+]>:$* $#ether $Q[$1] $:$2 especificacao numerica internet
```

```
R<Q[$+]>,$* $#ether $Q[$1] $:$2 especificacao numerica internet
```

```
R$*<Q[$+]> $#ether $Q[$2] $:$1 especificacao numerica internet
```

```

# entrega a hospedeiros ethernet conhecidos explicitamente especificados no
# nosso dominio
R$*<@%y.LOCAL>$* $#ether $0$2 $:$1<0$2>$3 usuario@hospedeiro.sun.com

# hospedeiro_ether.uucp e' tratado como hospedeiro_ether.$m aqui.
# Isso permite denominar de um local uucp foo!sun!hospedeiro_ether!usuario
R$*<@%y.uucp>$* $#ether $0$2 $:$1<0$2>$3 usuario@hospedeiro_ether.uucp

# Explicitamente especifica nomes no nosso dominio --
# os quais os nomes de hospedeiros nao sao conhecidos
R$*<@%*.LOCAL>$* $#error $:Never heard of host $2 in domain $m

# Limpa as denominacoes para uso externo -- retira LOCAL, encaminhamento ,=>:
R$* $:$>9 $1 entao continua...

# Passa os nomes validos ao hospedeiro retransmissor
R$*<@%*.+>$* $#M $0$R $:$1<0$2.$3>$4 usuario@qualquer.dominio

# Todas as denominacoes nas regras acima sao absolutas
# (qualificada completamente com dominios)
# As denominacoes abaixo podem ser parcialmente qualificadas
# entrega a hospedeiros ethernet conhecidos
R$*<@%y>$* $#ether $0$2 $:$1<0$2>$3 usuario@hospedeiro_ether

# outros nomes nao locais que nao tem para onde ir; voltam ao remetente
R$*<@%+.$->$* $#error $:Unknown domain $3
R$*<@%+>$* $#error $:Never heard of $2 in domain $m
R$*@%* $#error $:I don't understand $10$2

# Denominacoes locais com % nao sao realmente locais
R$+%$+ $0$>30$10$2 troca "%" por "@", e tenta de novo

# tudo o que sobra e' nome local
R$+ $#local $:$1 nomes locais

```


Apêndice G

Glossário

Análise - Parsing

Armazena-e-Remete - Store-and-forward

Armazenamento temporário - Buffering

Assíncrona - Start-Stop

BITNET - Because It's Time Network

BM - interface de usuário para correio no KA9Q

CCITT - International Telegraph and Telephone Consultative Comitee

CFC - software de apoio ao Sendmail. traduz do formato do **sendmail.cf** para uma linguagem de mais alto nível.

CLH - Clearinghouse - programa gerenciador de endereços de nó na Intergraph

Clix - Unix System V da Interpro

Colocar em Rede - Networking

Conjunto relacionado a - Suite

CTS - Clear To Send - sinal de interface serial

daemon - processo ativo em *background*

DCL - Digital Command Language - processador de comandos do VMS

DEC - Digital Equipment Corporation

DECSERVER - servidor de terminais da Digital

DECUS UUCP - Implementação de UUCP fornecida pela DECUS

Despachador de Correspondência - Mailer

Dígito de Verificação - Checksum

- DLM** - Data Link Mapping - conceito da DECNET para permitir a utilização de circuito virtual X.25
- DNS** - Domain Name Service - Serviço para prover determinação dinâmica de nome-endereço (sistemas Sun)
- DoD** - Department of Defense
- DTR** - Data Terminal Ready - sinal de interface serial
- EASE** - software de apoio ao Sendmail. Traduz de uma linguagem de mais alto nível para o formato do `sendmail.cf`
- ECD (Equipamento de Terminação de Circuito de dados)** - DCE (Data Circuit Terminating Equipment)
- ELM** - software público para interface de usuário para correio eletrônico
- Entroncamento** - Gateway
- Encaminhada** - Routed
- ETD** - Equipamento Terminal de Dados - DTE (Data Terminal Equipment)
- Ethernet** - Abordagem de rede local
- FTP** - File Transfer Protocol (protocolo de transferência de arquivo)
- Fusion** - Implementação TCP/IP da Network Research Corporation para VMS
- GNU** - GNU's Not Unix. Projeto de uma fundação de software livre
- Hop** - salto
- ICMP** - Internet Control Message Protocol (protocolo de controle de mensagens Internet)
- INC** - Intergraph Network Core - pacote de rede da Intergraph
- IP** - Internet Protocol
- KA9Q** - Implementação TCP/IP para microcomputadores
- LAN** - Local Area Network (rede local)
- MCR** - Modem Control Register - registrador de controle de modem
- Modem** - Modulador/demodulador
- MSR** - Modem Status Register - Registrador de estado de modem
- MTA** - Message Transfer Agent - agente de transferência de mensagem
- NCP** - Network Control Program - Programa que permite controlar e monitorar componentes do DECNET

- NIC** - Network Information Center - Centralizador de informações para a Internet
- NFS** - Network File Service - permite o uso de disco remoto de modo transparente (como diretório local)
- NIS** - Network Information Service - Pacote da Sun que provê compartilhamento de bases administrativas
- OSI** - Open System Interconnection (interconexão de sistemas abertos)
- PAD** - Packet Assembly-Disassembly (montagem-desmontagem de pacotes)
- PMDF** - Pascal Memo Distribution File - despachador de correspondência usado no VMS
- PSDN** - Packet Switch Data Network - rede de comutação de pacotes
- PSI** - Packetnet System Interface. Software que permite acesso a PSDN
- RENPAc** - Rede Nacional de Pacotes
- reply** - resposta a correspondência
- RNP** - Rede Nacional de Pacotes
- RTS** - Request To Send - sinal de interface serial
- Sendmail** - Despachador de correspondência. Também implementa SMTP
- SMTP** - Simple Mail Transfer Protocol (protocolo de transferência de correspondência simples)
- SRI** - Stanford Research Institute
- Submissão para remessa e entrega** - Submit
- SunOS v4.1** - Unix BSD para estações Sun
- TCP/IP** - Conjunto de protocolos usados na Internet
- UA** - User Agent - agente de usuário
- UDP** - User Datagram Protocol (protocolo de datagrama de usuário)
- UIP** - User Interface Presentation
- UUCP** - Unix to Unix CoPy
- UUPC** - implementação de UUCP para micros
- Varredura** - Polling
- WAN** - Wide Area Network (rede de área ampla)
- X.3** - Define a facilidade de empacotar/desempacotar numa rede de dados pública

- X.25** - Define os protocolos que atravessam o limite entre um nó usuário (ETD), e um nó de rede comutada (ECD)
- X.28** - Define os protocolos entre um ETD modo caracter e um PAD
- X.29** - Define os protocolos entre um PAD e um ETD modo pacote
- X.400** - padrão CCITT para correio eletrônico
- XNS** - Xerox Network System

Bibliografia

- [BE83] Eric Benhamou and Judy Estrin. Multilevel internetworking gateways: Architecture and applications. *Computer*, pages 27–34, September 1983.
- [Bur83] F.M. Burg. Design consideration for using the x.25 packet layer in data terminal equipment. In *Proceedings of INFOCOM 83*, pages 180–188, 1983. in TUTORIAL: COMPUTER COMMUNICATIONS : Architectures, Protocols, and Standards, Second Edition, William Stallings, IEEE Computer Society Press, 1987.
- [CK78] Vinton G. Cerf and Peter T. Kirstein. Issues in packet-network interconnection. *Proceedings of IEEE*, pages 180–202, November 1978. in TUTORIAL: PRINCIPLES OF COMMUNICATION AND NETWORKING PROTOCOLS, Simon S. Lam, IEEE Computer Society Press, 1984.
- [Cof88] Stephen Coffin. *UNIX: The complete reference*. Osborne McGraw-Hill, 1988.
- [Com88] Douglas Comer. *Internetworking with TCP/IP. Principles, protocols and architecture*. Prentice-Hall, Inc., 1988.
- [Con83] J.W. Conard. Services and protocols of the data link layer. *Proceedings of the IEEE*, pages 1378–1383, December 1983. in TUTORIAL: COMPUTER COMMUNICATIONS : Architectures, Protocols, and Standards, Second Edition, William Stallings, IEEE Computer Society Press, 1987.
- [Cun83] Ian Cunningham. Message-handling systems and protocols. *Proceedings of the IEEE*, 71(12):1425–1430, December 1983.
- [Cun84] Ian Cunningham. Electronic mail standards to get rubber-stamped and go worldwide. *Data Communications*, pages 159–168, May 1984.
- [DEC82] Digital Equipment Corporation. *Introduction to Local Area Networks*, 1982.
- [DeM87] Philip DeMar. Managing area numbers in multi-organizational Decnet networks. Recebido por mail, Dezembro 1987.
- [Dig88a] Digital Equipment Corporation. *VMS General user's manual*, Abril 1988.
- [Dig88b] Digital Equipment Corporation. *VMS System management, vol. 5a: Networking*, Abril 1988.
- [Dig89a] Digital Equipment Corporation. *Decserver 200 user's guide*, Junho 1989.

- [Dig89b] Digital Equipment Corporation. *Decserver management guide*, Junho 1989.
- [DK86] C.R. Dhas and V.K. Konangi. X.25: An interface to public packet networks. *IEEE Communications Magazine*, pages 18–25, September 1986. in TUTORIAL: COMPUTER COMMUNICATIONS : Architectures, Protocols, and Standards, Second Edition, William Stallings, IEEE Computer Society Press, 1987.
- [EMBa] Embratel. *Equipamentos Disponíveis para Comunicação de Dados pela Rede Telefônica na velocidade de 1200 BPS Assíncrono*.
- [EMBb] Embratel. *Ligação de Equipamentos à RENPAC*.
- [EMBc] Embratel. *Serviço RENPAC - Acesso Modo Caracter*.
- [EMBd] Embratel. *Serviço RENPAC 2000 - Manual de Utilização*.
- [EMBe] Embratel. *Serviço RENPAC 3025 - Equipamentos Conectáveis*.
- [EMBf] Embratel. *Serviço RENPAC 3028 - Manual de Utilização*.
- [EMBg] SISTEMA TELEBRÁS - Ministério das Comunicações. *Catálogo de Serviços*.
- [FV89] Ned Freed and Mark Vasoll. *PMDF v3.1 system manager's guide*. Innosoft International Inc., October 1989.
- [Gar88] Bdale Garbee. *The KA9Q Internet software package*, 1988.
- [Gui89] Fayez El Guibaly. Design and analysis of arbitration protocols. *IEEE Transactions on Computers*, 38(2), February 1989.
- [H⁺83] Robert Hinden et al. The darpa internet: Interconnecting heterogeneous computer networks with gateways. *Computer*, pages 38–48, September 1983.
- [HIP89] J. E. Hanrahan, Tom Allebrandi II. and Mark Pizzolato. *DECUS UUPC (VMSNET software) ver. 1.1 system manager's guide*. DECUS VAX systems SIG, Agosto 1989.
- [Int89] Intergraph Corporation. *CLIX TCP/IP user's guide*, Julho 1989.
- [Int90] Intergraph Corporation. *Intergraph network core user's manual*, Janeiro 1990.
- [IW87] J.E. Israel and A.J. Weissberger. Communicating between heterogeneous networks. *Data Communications*, pages 215–235, March 1987. in TUTORIAL: PRINCIPLES OF COMMUNICATION AND NETWORKING PROTOCOLS, Simon S. Lam, IEEE Computer Society Press, 1984.
- [Jac90] V. Jacobson. Compressing TCP/IP headers. Technical report, Network working group, 1990. RFC-1144.
- [Kil86] S.E. Kille. Mapping between x.400 and rfc 822. Technical report, University College London, June 1986. RFC-987.

- [L⁺85] B.M. Leiner et al. The darpa internet protocol suite. *IEEE Communications Magazine*, pages 29–34, March 1985. in TUTORIAL: COMPUTER COMMUNICATIONS : Architectures, Protocols, and Standards, Second Edition, Willian Stallings, IEEE Computer Society Press, 1987.
- [Lam80] Simon S. Lam. A carrier sense multiple access protocol for local networks. *Computer Networks*, 4:21–32, 1980. in TUTORIAL: PRINCIPLES OF COMMUNICATION AND NETWORKING PROTOCOLS, Simon S. Lam, IEEE Computer Society Press, 1984.
- [Lam83] Simon S. Lam. Multiple access protocols. *Computer Communications*, Vol. 1: Principles, W.Chou (editor), 1983. in TUTORIAL: PRINCIPLES OF COMMUNICATION AND NETWORKING PROTOCOLS, Simon S. Lam, IEEE Computer Society Press, 1984.
- [Lam84] Simon S. Lam. Data link control protocols. In S.S. Lam, editor, in *TUTORIAL: PRINCIPLES OF COMMUNICATION AND NETWORKING PROTOCOLS*, chapter 2. IEEE Computer Society Press, 1984.
- [LJF86] Lawrence H. Landweber, Dennis M. Jennings, and Ira Fuchs. Research computer networks and their interconnection. *IEEE Communications Magazine*, 24(6):5–16, June 1986.
- [Lot87] M. Lottor. Domain administrators operations guide. Technical report, SRI International, November 1987. RFC-1033.
- [MB76] Robert M. Metcalfe and David R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Communications of the ACM*, pages 395–404, July 1976. in TUTORIAL: LOCAL COMPUTER NETWORKS, SECOND EDITION, Keneth J. Thurber and Harvey A. Freeman, IEEE Computer Society Press, 1981.
- [Moc87a] P. Mockapetris. Domain names - concepts and facilities. Technical report, USC Information Science Institute, 1987. RFC-1034.
- [Moc87b] P. Mockapetris. Domain names - implementation and specification. Technical report, USC Information Science Institute, 1987. RFC-1035.
- [Net88] Network Research Corporation. *Fusion Network Software - VMS Programmer's reference manual*. Setembro 1988.
- [Net89a] Network Research Corporation. *Fusion Network Software - Network administrator's manual*. Outubro 1989.
- [Net89b] Network Research Corporation. *Fusion Network Software - VMS User's manual*. Outubro 1989.
- [NS82] Allen Newell and Robert F. Sproull. Computer networks: Prospects for scientists. *SCIENCE*, 215:843–851, February 1982.
- [Ott87] Tony Ottery. Flexibility the key with x400. *Computer Bulletin*, March 1987.

- [Pad83] M.A. Padlipsky. A perspective on the arpanet reference model. In *Proceedings of INFOCOM 83*, pages 242–253, 1983. in TUTORIAL: COMPUTER COMMUNICATIONS : Architectures, Protocols, and Standards, Second Edition, William Stallings, IEEE Computer Society Press, 1987.
- [Par86] Craig Partridge. Mail routing and the domain system. Technical report, CSNET CIC BBN Laboratories Inc., January 1986. RFC-974.
- [Pos82] J. Postel. Simple mail transfer protocol. Technical report, USC Information Science Institute, 1982. RFC-821.
- [QH86] John S. Quarterman and Josiah C. Hoskins. Notable computer networks. *Communications of the ACM*, 29(10):932–970, October 1986.
- [R⁺86] W.F. Racke et al. Profs/x.400 at cebit 86. Technical Report 8605, ENC (European Networking Center), 1986. IBM - Europäisches Zentrum für Netzwerkforschung.
- [Rob78] Lawrence G. Roberts. A evolution of packet switching. *Proceedings of IEEE*, 1978. in TUTORIAL: DISTRIBUTED PROCESSOR COMMUNICATION ARCHITECTURE, Kenneth J. Thurber, IEEE Computer Society Press, 1979.
- [Rom90] J. Romkey. A non-standard for transmission of IP datagrams over serial lines: SLIP. Technical report, Network working group, 1990. RFC-1055.
- [RW83] David D. Redell and James E. White. Interconnecting electronic mail systems. *Computer*, pages 55–63, September 1983.
- [Ryb80] A. Rybczynski. X.25 interface and end-to-end virtual circuit service characteristics. *IEEE Transactions on Communications*, pages 500–510, April 1980. in TUTORIAL: PRINCIPLES OF COMMUNICATION AND NETWORKING PROTOCOLS, Simon S. Lam, IEEE Computer Society Press, 1984.
- [S⁺82] J.F. Shoch et al. Evolution of the ethernet local computer network. *Computer*, pages 10–26, August 1982. in TUTORIAL: PRINCIPLES OF COMMUNICATION AND NETWORKING PROTOCOLS, Simon S. Lam, IEEE Computer Society Press, 1984.
- [Sch83] Norman F. Schneidewind. Interconnecting local networks to long-distance networks. *Computer*, pages 15–24, September 1983.
- [Sho78] J.F. Shoch. Inter-network naming, addressing, and routing. In *The Proceedings of the COMPCOM F78 Conference*, pages 72–79, 1978. in TUTORIAL: PRINCIPLES OF COMMUNICATION AND NETWORKING PROTOCOLS, Simon S. Lam, IEEE Computer Society Press, 1984.
- [Sta84] William Stallings. *Local Networks*. Macmillan Publishing Company, 1984.
- [Sta87] M. Stahl. Domain administrators guide. Technical report, SRI International, November 1987. RFC-1032.

- [Sun90a] Sun Microsystems, Inc. *SunOS system and network administration*, March 1990.
- [Sun90b] Sun Microsystems, Inc. *SunOS v4.1 reference manual*, March 1990.
- [Tra90] Mônica M. R. G. Travieso. Aspectos de princípios e nomenclatura, modelos e padrões e implementações para viabilização de correio eletrônico. Technical report, Instituto de Ciências Matemáticas de São Carlos, Maio 1990.
- [Tru] Dave Trulli. *BM user's manual*.
- [Uns82] M.B. Unsoy. X.75 internetworking of datapac with other packet switched networks. *Journal of Telecommunication Networks*, pages 239–251, 1982. in TUTORIAL: PRINCIPLES OF COMMUNICATION AND NETWORKING PROTOCOLS, Simon S. Lam, IEEE Computer Society Press, 1984.
- [Wai87] Mitchell Waite, editor. *Unix Papers for Unix developer and power users*. Howard W. Sams and Co., 1987.
- [Wei83] A.J. Weissberger. Bit-oriented data link controls. *Computer Design*, pages 195–206, March 1983. in TUTORIAL: COMPUTER COMMUNICATIONS : Architectures, Protocols, and Standards, Second Edition, William Stallings, IEEE Computer Society Press, 1987.