UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

Departamento de Informática

Técnicas para aumento da capacidade de escala em sistemas de publicação e subscrição de informação

Mário Luís de Jesus Rodrigues Guimarães

(Licenciado)

Dissertação para obtenção do grau de

Mestre em Informática

Fevereiro de 2002

Técnicas para aumento da capacidade de escala em sistemas de publicação e subscrição de informação

Mário Luís de Jesus Rodrigues Guimarães

Dissertação submetida para provas

de mestrado em

Informática

Departamento de Informática

Faculdade de Ciências da Universidade de Lisboa

Lisboa

Fevereiro de 2002

Dissertação realizada sob a orientação do

Doutor Luís Eduardo Teixeira Rodrigues

Professor Associado

Faculdade de Ciências da Universidade de Lisboa

Resumo

Neste trabalho, começamos por abordar o paradigma de *Publicação e Subscrição de Informação* e apresentamos algumas das suas concretizações mais relevantes. Em seguida, introduzimos um conjunto de técnicas capazes de aumentar a capacidade de escala em sistemas baseados neste paradigma de comunicação, quando utilizados em redes alargadas (*WAN*). Assim, propomos a criação de um modelo de espaço de informação baseado numa topologia de *domínios de publicação*. Esta organização da rede em domínios permite introduzir o conceito de *subscrição-orientada* ao domínio. Segundo este conceito, somente os editores dentro de um domínio podem publicar para os assuntos deste. Como tal, os pedidos de subscrição são dirigidos para os domínios fonte da informação referenciada, evitando a sua dispersão por toda a rede.

A arquitectura proposta neste trabalho assenta num modelo em rede de nós distribuidores de eventos, e recorre aos mecanismos de difusão em grupos *IP Multicast* para entrega de notificações aos subscritores locais a um nó. Surge assim a necessidade de desenvolver um algoritmo que execute o emparelhamento eficiente de expressões de interesse em grupos de difusão. Atendendo à dimensão do espaço de hipóteses de emparelhamento, o uso de técnicas exaustivas de pesquisa revela-se impraticável. Assim, na segunda metade desta tese, iremo-nos concentrar na apresentação de um algoritmo de procura genética, e mostramos resultados duma simulação do mesmo.

Entendemos que as técnicas abordadas permitem a criação de sistemas de publica-

ção e subscrição, mais eficientes, seguros e capazes de escalar. Esta dissertação pretende assim contribuir para o desenvolvimento de serviços de difusão de informação na *Internet*, assentes no paradigma de publicação e subscrição de informação.

Abstract

In this work we present the *Publish and Subscribe* communication paradigm and some relevant implementations. We also propose a number of technics so this type of systems can scale in wide area networks (WAN). We begin proposing an information space model based on a topology of *publishing domains*. This network organization introduces the notion of *oriented-subscription*. This concept states that only publishers inside a domain can publish to the domain's subjects, and so, subscription requests are directed to source domains avoiding their dissemination throughout the network.

The architecture proposed in this work uses a network of event nodes and supports IP Multicast when delivering notifications to local subscribers at a node. Following this, we have to develop an algorithm that efficiently maps expressions of interest into multicast groups. Considering the search space dimension of possible mappings, the use of an exhaustive search technic reveals to be unfeasible. As such, on the second half of this thesis we present a genetic search algorithm and evaluate this algorithm using simulations.

We believe the proposed technics support the development of more efficient, secure and scalable *Publish and Subscribe* systems.

Palavras Chave

Publicação/Subscrição

Difusão *IP*

Protocolos de comunicação

Sistemas Distribuídos

Algoritmos Genéticos

Agrupamento de Subscrições

Keywords

Publish/Subscribe

IP Multicast

Communication Protocols

Distributed Systems

Genetic Algorithms

Subscription Mapping

Agradecimentos

Agradeço ao Professor Doutor Luís Rodrigues pelo seu empenho na orientação desta dissertação, e pelas suas úteis sugestões. Agradeço também à Professora Doutora Graça Gaspar pelos comentários relativos ao algoritmo de pesquisa genética.
Muito obrigado a ambos.

Alcochete, Fevereiro de 2002
Mário Luís de Jesus Rodrigues Guimarães

À minha querida namorada, Gabriela, pela sua compreensão e apoio

Aos meus queridos pais, Mário e Julieta, pelo seu apoio e dedicação

A Deus, pela força que me tem dado

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Nos últimos anos temos assistido a um crescimento enorme das redes de computadores, em particular da *Internet*, especialmente depois do advento da *World Wide Web*. Cada vez é maior o número de sistemas distribuídos que interligados, criam, transformam e consomem informação. Sobre estes sistemas interligados têm sido colocadas grandes exigências quanto à sua capacidade evolutiva e adaptativa, focando na facilidade de integrar e transformar informação.

Neste contexto em que a informação assume um papel central, o paradigma de publicação e subscrição de informação (do inglês *publish and subscribe*) [1, 2] tem criado um interesse crescente em virtude das possibilidades que introduz no campo da computação distribuída. Ao contrário da comunicação ponto-a-ponto[1], onde cada processo do par comunicante tem obrigatoriamente de conhecer a identidade ou a localização do seu interlocutor, o paradigma de publicação e subscrição permite que a comunicação seja realizada ao mesmo tempo entre dois ou mais processos de forma distribuída e anónima. Estas propriedades devem-se ao facto de não existirem quaisquer ligações explícitas entre componentes do sistema, desconhecendo cada uma destas a existência das

[1] Da qual são exemplos os protocolos *TCP* e *UDP* [3], ou as chamadas a procedimentos remotos [4, 5].

restantes. Consequentemente cabe à infra-estrutura que realiza este paradigma de comunicação, servir de intermediária na troca de informação entre os diversos interlocutores. Assim, neste paradigma, a informação constitui a única interface de ligação entre as múltiplas componentes dum sistema distribuído. As aplicações passam a consumir informação de acordo com os interesses manifestados perante o sistema, desconhecendo quem a produz.

Este trabalho propõe uma nova adaptação do paradigma de publicação e subscrição de informação. Esta modificação é necessária de forma a tornar exequível a criação de um serviço de publicação e subscrição na *Internet*, à semelhança da escala de outros serviços tais como a *World Wide Web* ou o *Email*. Consequentemente, pensamos que a realização de um serviço desta dimensão recorrendo às técnicas actualmente usadas [6, 7, 8, 9] não é viável, pois a força principal deste paradigma, a propriedade do anonimato da fonte de informação, reduz bastante a capacidade de escala de uma possível execução na *Internet*, e impossibilita a verificação da integridade da fonte de informação, tão importante quando se procura explorar serviços abertos à escala pretendida.

Como tal, propomos em primeiro lugar a definição de *domínios de publicação* na *Internet*, baseados em nomes *DNS* [10, 11] (*Domain Name System*) e geridos por uma autoridade de publicação no domínio. Seguidamente é introduzido o conceito de *subscrição-orientada* ao domínio. A criação de domínios de publicação aliada a este novo conceito de publicação e subscrição, permite alcançar níveis de escala superiores e desenvolver mecanismos eficientes de controlo de acesso à informação em cada domínio. Os subscritores continuam a desconhecer a identidade dos produtores de informação, mas no acto de subscrição passam a referir o nome do domínio de publicação fonte. Contudo, pensamos que esta parcial perda de anonimato da fonte de informação não reduz o potencial desta nova solução. É o custo que temos de assumir em virtude do aumento da capacidade de escala e da possibilidade de autenticação da fonte de informação.

Estes conceitos estão na base do desenvolvimento de uma nova arquitectura para sistemas de publicação e subscrição de informação, que apresentamos. Esta arquitectura é formada por uma topologia em rede de nós distribuidores de eventos, e recorre à difusão local em grupos *IP Mulicast* [12], para distribuição eficiente da informação pelos subscritores. Assim, na segunda metade desta dissertação, aprofundamos um algoritmo de emparelhamento de subscrições em grupos de difusão, e apresentamos resultados duma simulação do mesmo.

Pensamos que as técnicas propostas permitem a criação de um sistema mais eficiente, seguro e capaz de escalar. Este trabalho pretende assim contribuir para o desenvolvimento de serviços de difusão de informação na *Internet*, assentes no paradigma de publicação e subscrição de informação.

O resto da dissertação está organizado da seguinte forma. No Capítulo dois, introduzimos o paradigma de publicação e subscrição de informação e apresentamos algumas realizações do mesmo, bem como as respectivas limitações na sua aplicação à escala da *Internet*. No Capítulo três, apresentamos um conjunto de técnicas para aumento da capacidade de escala e de segurança deste tipo de sistemas em redes abertas e de grande dimensão (*Wide Area Networks*). No Capítulo quatro aprofundamos uma destas técnicas, nomeadamente um algoritmo para agrupamento de subscrições em endereços de grupo *IP Multicast*, e no Capítulo cinco fazemos a sua avaliação, recorrendo a simulações. Por fim, no Capítulo seis concluímos este trabalho e abordamos vários pontos para futuros desenvolvimentos.

Uma primeira versão deste trabalho [13] foi apresentada na ”3ª Conferência sobre Redes de Computadores, CRC’2000”, Viseu, Novembro 2000.

Capítulo 2

Publicação e Subscrição de Informação

Na generalidade dos sistemas distribuídos, a forma de comunicação dominante assenta na troca de dados ponto-a-ponto (do inglês, *unicast*) entre dois processos. Nestes sistemas, as aplicações são desenhadas normalmente segundo o modelo pedido/resposta, mais conhecido por cliente/servidor. Contudo e apesar dos seus méritos, um sistema puramente cliente/servidor demonstra pouca flexibilidade e reduzida capacidade evolutiva, conduzindo a custos acrescidos de manutenção. Isto acontece porque os clientes estão dependentes da identidade do servidor (eventualmente, também da sua localização). Em outras aplicações, é mais adequado fazer com que cada cliente dependa somente da informação pretendida, independentemente de quem a produz.

Clarifiquemos esta ideia com um exemplo simples. Considere-se um sistema de informação duma unidade automatizada de fabrico de automóveis. Quando dá entrada uma nova encomenda, o serviço de encomendas pode enviar um pedido de material para o sistema de informação do armazém e também um aviso para o sector de produção para que possam ser preparadas as máquinas para a execução da encomenda. Considere-se também que é desenvolvido posteriormente um sistema de acompanhamento de encomendas de clientes, e é criado um armazém de dados (*Data Warehouse*) no qual é necessário integrar dados das encomendas. Se o sistema base tiver sido concretizado numa óptica cliente/servidor, o serviço de encomendas terá de ser alterado para passar a emitir avisos para cada um dos novos sistemas. Isto porque, numa arquitectura cliente/servidor é necessário identificar cada um dos servidores que necessitam da informação.

Por comparação, nos sistemas de publicação e subscrição de informação o papel de cliente é substituído pelo de *subscritor*, consumidor de informação, e o papel de servidor é substituído pelo de *editor*, produtor de informação. Editores enviam informação na forma de notificações de eventos para o sistema, que a distribui por quem a subscreve. Um *evento* traduz uma mudança de estado do sistema, enquanto que uma *notificação* é uma mensagem reportando a ocorrência de um evento[1]. Quem edita informação pode anunciar ao sistema qual é o tipo de informação que produz, enquanto que quem subscreve precisa de avisar o sistema a respeito do tipo de informação que deseja receber. Sempre que esta é publicada, o sistema é responsável por a entregar aos respectivos subscritores na forma de uma notificação.

A total ausência de ligação explícita entre editores e subscritores torna anónima a comunicação entre estes, sendo esta efectuada somente em função da informação pretendida. Deste modo, ganha-se independência de identidade entre emissores e subscritores, isto é, cada um deles não precisa de saber da existência de nenhum dos outros. Isto permite que possa variar o conjunto de editores e subscritores de cada fluxo de informação, sem que isso afecte o funcionamento do sistema.

Assim, o paradigma de publicação e subscrição de informação está para a composição de programas (*software*), assim como os barramentos electrónicos (*buses*) estão para a composição de equipamento (*hardware*), tornando possível adicionar e substituir trans-

[1] Apesar da sua distinção, neste documento os termos "evento" e "notificação" são por vezes utilizados em situações semelhantes. No entanto, a interpretação correcta deverá surgir pelo contexto em que se inserem.

parentemente componentes de código dum sistema distribuído, sem que isso implique obrigatoriamente a sua paragem mesmo que temporária, possibilitando o desenvolvimento de sistemas *"Plug'n' Play"*. A manutenção dos sistemas torna-se deste modo mais fácil, e as aplicações podem ser migradas ou replicadas, mesmo estando o sistema em exploração. Retomando o exemplo introdutório da fábrica de automóveis e do seu sistema de encomendas, se a arquitectura utilizada fosse de publicação e subscrição em vez do modelo cliente/servidor, não seria necessário alterar o serviço de entrada de encomendas sempre que surgisse um novo sistema interessado nesse evento. Numa arquitectura de publicação e subscrição de informação bastaria que aquando da entrada duma nova encomenda fosse emitida uma notificação para o sistema, que a encaminharia para os subscritores interessados.

2.1 Métodos de endereçamento

Sendo a informação a única ponte de ligação entre produtores e consumidores, é necessário algum mecanismo para a referenciar, tanto no acto de publicação como no de subscrição.

Actualmente, é possível identificar três técnicas principais de endereçamento da informação em sistemas de publicação e subscrição, a saber, as formas de endereçamento *por-canal* (do inglês *Channel-Based*), *por-assunto* (do inglês *Subject-Based*) e *por-conteúdo* (do inglês *Content-Based*).

2.1.1 Endereçamento *por-canal*

Pela sua simplicidade, o endereçamento *por-canal* é o método mais concretizado, estando na base dos sistemas que realizam a especificação do serviço de eventos da *CORBA, Common Object Request Broker Architecture*[14, 15]. Em arquitecturas *por-canal*, os subscritores lêem eventos de um canal endereçado através do seu identificador e os editores enviam notificações para o canal usando o mesmo identificador. Consoante o número de canais pretendidos, assim tantos pedidos de subscrição terão de ser efectuados.

2.1.2 Endereçamento *por-assunto*

O endereçamento *por-assunto* é a segunda forma de endereçamento mais utilizada pela generalidade dos sistemas de publicação e subscrição de informação [6, 16, 17, 18]. O endereçamento *por-assunto* recorre à noção de espaço de informação global e unidimensional, no qual a cada notificação está associado um único atributo designado por *assunto* (do inglês, *subject*), enquanto que o restante conteúdo da notificação, estruturado ou não, é ocultado ao sistema. Geralmente o assunto toma a forma de uma hierarquia de nomes (*hierarchical name space*), que, ao contrário de um formato plano (*flat name space*), permite classificar a informação, facilitando a sua selecção pelos subscritores pois os assuntos podem ser apresentados na forma de uma árvore facilmente pesquisável. Por exemplo, uma aplicação de visualização do estado dos activos financeiros numa bolsa poderá subscrever informação relativa ao comportamento bolsista das acções da XPTO, bastando para tal indicar que pretende receber informação associada ao assunto "/bolsa/acções/XPTO"[2], para o qual existe uma outra aplicação a publicar periodicamente notificações de alteração do seu estado. Numa outra situação, a aplicação poderia subscrever o estado de todas as acções especificando o assunto com caracteres de substituição (*wildcards*), isto é, indicando por exemplo ”/bolsa/acções/*”. Relativamente ao endereçamento *por-canal*, a principal diferença consiste na possibilidade de uma única subscrição poder endereçar um conjunto alargado de assuntos, mediante o uso de técnicas de emparelhamento de expressões com nomes de assunto.

[2]Neste texto o caracter ”/” é usado para denotar a hierarquia de nomes de assunto.

2.1.3 Endereçamento *por-conteúdo*

Por sua vez, os sistemas *por-conteúdo* definem um espaço de informação n-dimensional, introduzindo a possibilidade de filtrar a informação difundida, nas suas múltiplas dimensões [8, 9, 19, 20, 21, 22]. A informação é caracterizada por um conjunto de atributos a_0, a_1, ..., a_{n-1}, que permitem endereçar/seleccionar qualquer subconjunto de notificações do espaço global, em função da combinação lógica ($\wedge$ e $\vee$) de operações relacionais entre cada um dos atributos $a_0, a_1, \ldots, a_{n-1}$, e respectivos valores de comparação. Geralmente os operadores lógicos considerados são =, $\neq$, $<$, $>$, $\leq$ e $\geq$, tanto para valores escalares como para cadeias de caracteres. Algumas concretizações de endereçamento *por-conteúdo* possibilitam o uso de expressões regulares para comparação de cadeias de caracteres, embora essa funcionalidade possa reduzir o potencial de escala do sistema. Outras soluções, permitem que os editores possam especificar o espaço de informação que produzem dentro de um assunto, usando endereços *por-conteúdo* com um ou mais atributos condicionados a certos valores (SIENA [9]). Cada notificação poderá ainda conter outros dados para além dos seus atributos, não acessíveis ao sistema de publicação e subscrição. Esses dados, eventualmente estruturados, dizem apenas respeito à aplicação que faz uso do sistema. Voltando ao exemplo da bolsa, uma aplicação de supervisão do mercado poderia estar somente interessada em receber o estado duma acção quando o valor desta fosse maior que um certo limite. Em *por-conteúdo* isto poderia ser feito subscrevendo, por exemplo, o endereço (título = "/bolsa/acções/*", valor>$5.000\$00$).

2.1.4 Outras formas de endereçamento

Mais recentemente, o SIENA [9] introduz um novo método designado por endereçamento *por-padrão* (do inglês, *Pattern-Based*). Neste método é possível indicar ao sistema o interesse por uma determinada sequência de eventos. Assim, quando o sistema detecta a sequência especificada comunica a sua ocorrência aos subscritores interessados, através da geração de uma notificação.

O *TPS* (*Type-Based Publish/Subscribe*) [23] é uma variante de publicação e subscrição orientada por objectos. O *TPS* suporta os métodos de endereçamento *por-assunto* e *por-conteúdo* sobre estruturas de dados fortemente tipificadas, constituindo-se como um mecanismo ao nível da linguagem de programação.

Neste trabalho, iremos considerar apenas soluções que dão suporte aos endereçamentos *por-assunto* e *por-conteúdo*. O *por-canal* é uma forma simplificada do *por-assunto*, não sendo portanto relevante para este estudo.

2.2 Primitivas em Publicação e Subscrição

Todos os sistemas de publicação e subscrição de informação disponibilizam um conjunto de funções (*Application Programming Interface*), a utilizar pelo conjunto de editores e subscritores comunicantes. Independentemente do sistema, todos exportam três funções essenciais,

- *publicar()*, uma notificação para o sistema;
- *subscrever()*, um conjunto de notificações que satisfaça uma determinada condição de filtro. No caso do endereçamento *por-canal* o filtro será o identificador do canal, em *por-assunto* representa uma expressão de emparelhamento com nomes de assunto, e no caso do endereçamento *por-conteúdo* uma condição sobre os atributos do sistema de eventos;
- *cancelar()*, uma subscrição anteriormente submetida;

Para além destas funções básicas, cada sistema poderá disponibilizar funcionalidades adicionais. Estas poderão estar relacionadas, por exemplo, com qualidades de serviço (*QoS*) oferecidas (TIB/Rendezvous [6]), extensões de suporte à mobilidade de componentes (SIENA [9]), ou com características particulares do desenho do sistema.

2.3 Concretizações do paradigma

Actualmente, apesar da multiplicidade de soluções para o paradigma de publicação e subscrição de informação, evidenciam-se dois modelos de arquitectura fundamentais, nomeadamente, o modelo de nós distribuidores[3] e o modelo de difusão.

2.3.1 Modelo de rede de nós

No primeiro caso, existe uma rede de nós distribuidores ligados entre si, de modo hierárquico, acíclico ou livre, consoante a complexidade e o desenho do sistema. A cada um destes distribuidores poderão estar ligados editores e subscritores de informação, que em conjunto com os primeiros completam o sistema de eventos.

Como se pode observar na Fig. 2.1, editores e subscritores nunca trocam notificações directamente. Todas as notificações fluem dos primeiros para os últimos através da rede de distribuidores de acordo com algum algoritmo de encaminhamento, adaptado a cada uma das topologias. Geralmente, a informação de encaminhamento em cada nó deriva da propagação dos pedidos de subscrição pela rede, desde os subscritores até aos distribuidores ligados aos editores da informação requerida.

A disseminação total de subscrições é necessária se e só se a informação de encaminhamento em cada nó resultar apenas das subscrições e nada mais. Alguns sistemas

[3]"Nós distribuidores", "nós servidores", "distribuidores", "servidores", ou simplesmente "nós".

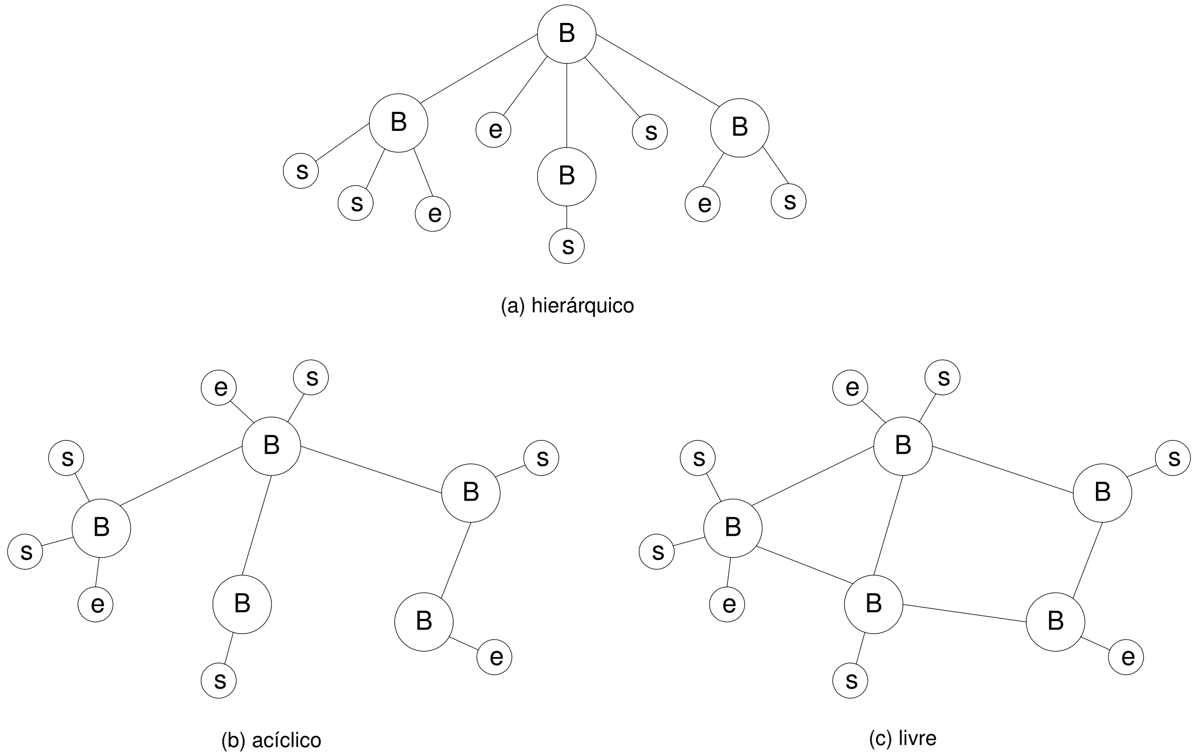


Figura 2.1: Topologias de sistema de eventos com nós distribuidores.

utilizam também mensagens de anúncio de publicação, emitidas pelos editores de informação, de maneira a optimizar a propagação dos pedidos de subscrição pela rede (SIENA [9]). Estas, sendo difundidas em toda a rede permitem que mais tarde não seja necessário dispersar totalmente os pedidos de subscrição, bastando dirigi-los para os distribuidores próximos das fontes de informação que fizeram anúncios correspondentes. Este mecanismo usa as mensagens de anúncio para estabelecer o caminho mais curto das subscrições até aos editores em causa, e estas por sua vez estabelecem o caminho inverso para as notificações.

A capacidade de retenção de conhecimento das subscrições em cada distribuidor, permite mais facilmente a estes sistemas realizar o endereçamento *por-conteúdo*. Num sistema que dê suporte a esta forma de endereçamento, cada distribuidor inspecciona o conteúdo de cada notificação recebida e entrega-a de acordo com os interesses manifestados pelos seus vizinhos, sejam estes outros distribuidores ou simplesmente subscritores.

Entre os vários sistemas existentes que realizam o modelo de rede de nós (Elvin [8], SIENA [9], Keryx [20], Gryphon [24]), está a popular rede *USENET* de servidores *NNTP* (*Network News Transport Protocol*)[17]. Nesta rede os servidores são organizados hierarquicamente, formado relações mestre/escravo entre si. Os servidores *NNTP* trocam informação na forma de artigos sobre determinados tópicos, sendo possível indicar simultaneamente o interesse por um conjunto de tópicos recorrendo a expressões simples de emparelhamento, como por exemplo, "comp.lang.*". Contudo, um dos problemas com este sistema consiste na limitação dos seus mecanismos de filtragem de artigos. De facto, são várias as aplicações cliente de acesso ao serviço de *News* que realizam mecanismos sofisticados para filtragem de mensagens, após estas terem sido descarregadas a partir do servidor *NNTP*. Esta limitação, provoca a troca de grandes quantidades de informação entre servidores, que posteriormente não é do interesse dos utilizadores do serviço. De facto, Levin et al.[25] concluem que se os servidores recebessem apenas artigos de tópicos para os quais tivesse havido um interesse recente, a redução das necessidades de armazenamento e de largura de banda seria de mais de 80%.

2.3.2 Modelo de difusão

Neste modelo a distribuição de notificações está totalmente dependente dos protocolos de difusão utilizados ao nível da rede, não existindo qualquer rede de nós distribuidores intermédios entre editores e subscritores, pelo que a introdução eficiente de mecanismos de filtragem *por-conteúdo* se torna difícil. Assim, estes sistemas concretizam apenas o endereçamento *por-assunto*, sendo menos complexos que os anteriores.

O IONA's OrbixTalk[7, 16] é um exemplo de um sistema com endereçamento *por-assunto* realizado sobre um protocolo de difusão, o *IP Multicast*. Este sistema faz a tradução de endereços *por-assunto* em endereços de grupo *IP Multicast* com base num directório de nomes de assuntos centralizado, que assume a responsabilidade de atribuir um grupo livre sempre que seja criado um assunto novo. Quando chega um pedido de subscrição de um cliente para um assunto existente, o serviço responde-lhe com o endereço do grupo atribuído, necessário para que esse subscritor se torne membro e passe a receber as notificações correspondentes ao assunto endereçado. No OrbixTalk, o algoritmo de atribuição de endereços *IP Multicast* funciona de forma cíclica quando o número de assuntos a emparelhar excede o número de endereços de grupo disponíveis, fazendo multiplicar o número de assuntos em cada grupo de difusão. Consequentemente, cada assunto possui um identificador numérico de forma a tornar mais célere a filtragem das notificações recebidas por um subscritor.

Outro exemplo de sistema de publicação e subscrição de informação baseado em tecnologias de difusão, é o TIBCO's TIB/Rendezvous[6]. O TIB/Rendezvous, funciona em difusão total (do inglês, *broadcast*) em segmentos de rede local[4]. Cada máquina que participa no sistema corre um módulo de código que escuta todo o tráfego difundido, procurando mensagens que pertençam a assuntos subscrevidos pelas aplicações locais. Mais uma vez é realizado o endereçamento *por-assunto*. O TIB/Rendezvous pode ser utilizado em redes alargadas (*WAN*) sendo necessário configurar manualmente os parâmetros de encaminhamento entre as redes, quer estas sejam geograficamente próximas ou remotas. Neste processo de conexão são utilizados módulos de código especiais (designados por *routing daemons*) que executam o encaminhamento das notificações pelas redes ligadas.

[4]No TIB/Rendezvous o uso do *IP Multicast* é sugerido como uma forma de optimizar a filtragem de notificações recorrendo à selecção ao nível da rede.

2.3.3 Qualidade de Serviço

Qualquer sistema real implementa algumas propriedades adicionais à sua funcionalidade básica, necessárias ao bom funcionamento do sistema durante a realização da sua missão. No contexto de um serviço de eventos, entende-se a *Qualidade de Serviço* por um conjunto de propriedades não funcionais, que não afectando a semântica do serviço de eventos, são de grande importância na sua realização e utilização [9]. Estas propriedades não-funcionais caracterizam o serviço de eventos e podem respeitar, entre outros,

- à segurança: autenticação de editores e subscritores; integridade e confidencialidade da informação trocada;
- às propriedades de entrega: fiável; garantida; transaccional;
- à capacidade de tempo-real: garantias temporais na entrega de notificações; prioridades;
- à memória: tamanho dos tampões internos (*buffers*) e das filas de espera (*queues*);
- ao balanceamento de carga.

Refira-se que a generalidade dos sistemas de publicação e subscrição de informação comerciais oferecem diferentes graus de qualidade de serviço. Destes, salientamos os vários níveis de entrega de notificações disponíveis [7, 26]. A entrega é *fiável*, quando apesar de problemas transitórios no sistema as notificações são entregues aos subscritores (à semelhança do protocolo *TCP*). A entrega de notificações é *garantida*, quando apesar dos processos subscritores estarem sujeitos a falhas e reinicializações, estes as recebem após recuperação. A garantia é dada desde que se respeitem certos limites temporais e de armazenamento impostos pela configuração do sistema. Por último, diz-se que a entrega é *transaccional*, quando o sistema assegura que todos os subscritores dum conjunto recebem uma sequência de notificações, ou que nenhum recebe em caso de falha na entrega a um deles.

2.4 Limitações

2.4.1 Limitações dos métodos de endereçamento

Qualquer dos métodos de endereçamento apresentados não possui a capacidade de especificar uma região para difusão delimitada de notificações. Ou seja, não é possível um editor emitir uma notificação e indicar uma zona limite para a sua propagação. Note-se que esta limitação é independente do modelo de arquitectura utilizado.

A falta desta capacidade de definição de zonas delimitadas de difusão nos actuais métodos de endereçamento, faz com que numa instalação em rede privada se tenham de introduzir modificações à arquitectura original, que por exemplo, isolem por defeito o sistema do exterior mas que dêem suporte à exportação explícita de assuntos para fora da rede, rejeitando-se qualquer pedido de subscrição vindo do exterior para um assunto não exportado. Por exemplo, uma empresa poderá pretender difundir para o seu exterior notícias sobre os seus produtos, mas certamente não quererá deixar passar informação privilegiada que circule pelo sistema de publicação interno. Geralmente, existem mecanismos ao nível da rede que permitem delimitar a propagação dos pacotes. Por exemplo, em *IP* é possível recorrer à configuração do campo *TTL* (*Time-To-Live*) de cada pacote.[5]

Convém referir que nenhum dos sistemas analisados apresenta esta capacidade.

[5]Mais recentemente, a *Internet Multicast Address Allocation Architecture* [27] define formas mais eficientes de circunscrição de pacotes a regiões da rede do que a utilização do campo *Time-To-Live* (ver Apêndice A)

2.4.2 Limitações das concretizações existentes

2.4.2.1 Modelo de rede de nós

Apesar das vantagens do modelo de rede de nós distribuidores, particularmente a sua capacidade de retenção distribuída de conhecimento das subscrições e de suporte a sistemas *por-conteúdo*, esta solução apresenta várias limitações.

A primeira reside no dilema entre a expressividade ao nível do endereçamento *por-conteúdo*, isto é, a riqueza dos operadores disponíveis para selecção do conteúdo, e a capacidade de escala do sistema. Em relação a este ponto, convém relembrar que nestes sistemas todos os distribuidores realizam tarefas de filtragem e encaminhamento das notificações recebidas, e como tal, quanto maior for o peso da execução das operações de filtragem menor será a capacidade de atender a um número crescente de eventos dentro do sistema.

A segunda limitação está relacionada com os mecanismos de encaminhamento dos pedidos de subscrição. Em geral a solução passa normalmente pela disseminação total das subscrições para todos os distribuidores do sistema, uma vez que qualquer emissor ligado a qualquer um dos distribuidores pode publicar para qualquer assunto, pois não existem restrições na publicação em função da localização da fonte de informação.

No entanto, como vimos, é possível evitar a necessidade da propagação total de subscrições, através da utilização de mensagens de anúncio de novas publicações. Esta técnica reduz efectivamente o número de mensagens de subscrição trocadas entre os distribuidores do sistema à custa da dispersão total das mensagens de anúncio. Como estas são em princípio em número muito mais reduzido do que os pedidos de subscrição, sendo geralmente emitidas uma única vez por cada nova fonte de um assunto, o saldo final deste mecanismo é positivo. Ainda assim, num sistema de grandes dimensões (por exemplo em *WAN*) onde a quantidade de assuntos pode ser na ordem dos milhares ou mais, o custo para os distribuidores do conhecimento de todas as fontes de informação pode ser significativo.

Uma terceira desvantagem deste modelo, reside no facto da generalidade dos sistemas nele baseados não fazerem uso dos avanços na área da difusão de informação na *Internet*, nomeadamente os relacionados com o *IP Multicast*. Em vários sistemas a entrega de notificações aos subscritores locais a um nó é feita ponto-a-ponto e não por difusão, pelo que a sua capacidade de escala se reduz quando múltiplas cópias de uma notificação têm de ser enviadas a um número crescente de subscritores. Se a entrega fosse efectuada por difusão, um único pacote seria suficiente.

2.4.2.2 Modelo de difusão

Os sistemas de publicação e subscrição de informação realizados segundo este modelo tiram partido imediato da evolução dos protocolos de difusão na *Internet*, nomeadamente do *IP Multicast*, o que lhes dá algumas vantagens em relação aos sistemas anteriores. Por exemplo, nestes sistemas é possível efectuar difusões circunscritas a regiões da rede. Também é possível endereçar um grupo de subscritores e entregar uma notificação ao grupo numa única mensagem, poupando-se recursos da rede. Contudo, embora a sua capacidade de escala seja maior, estas soluções têm concretizado apenas o endereçamento *por-assunto*, principalmente devido à simplicidade e reduzida expressividade dos mecanismos de endereçamento ao nível da rede (por exemplo em *IP Multicast* ou se subscreve o grupo todo ou não se subscreve nada).

Outro problema que estas soluções têm, reside na forma como realizam o emparelhamento de assuntos em grupos, estando a sua qualidade muito dependente da eficiência do algoritmo utilizado. Geralmente, a cada assunto é atribuído um grupo retirado de uma lista de grupos disponíveis. Contudo, numa solução destas, para um número muito elevado de assuntos é preciso uma igual quantidade de grupos. Observando o estado tecnológico actual, esta solução pode constituir uma situação incomportável para as tabelas de encaminhamento ao nível da rede, resultando em sobrecarga e degradação do sistema.

Normalmente, a solução passa por alguma forma de reutilização de grupos. Por exemplo, o OrbixTalk pretende responder a este problema através da limitação do número de grupos atribuíveis. Esta limitação pode resultar a partir de certo momento no esgotamento dos endereços *IP Multicast* livres, face a uma quantidade maior de assuntos por emparelhar. Quando isto acontece, o algoritmo distribui vários assuntos num único grupo *IP Multicast*. Embora esta técnica ajude a resolver o problema de sobrecarga das tabelas de encaminhamento, outro é criado, pois vários subscritores podem receber informação que não pretendiam.

Quanto ao TIB/Rendezvous, a difusão total das notificações implica necessariamente que qualquer máquina pode receber eventos que não lhe interessa, sendo portanto pouco eficiente relativamente aos recursos computacionais e de rede consumidos, em particular quando o número de subscritores interessados numa notificação for uma percentagem reduzida do total dos subscritores. Um problema de escala surge quando se pretende interligar sistemas remotos recorrendo às componentes de código disponibilizadas. Nesta situação é estabelecida manualmente uma ligação ponto-a-ponto persistente em forma de túnel (*tunneling*) entre duas redes remotas, possibilitando a exportação e importação de assuntos entre ambas. Neste caso, qualquer mensagem de um assunto exportado é difundida totalmente por todas as redes remotas directamente ligadas, que tenham importado esse assunto. Ou seja, a difusão total continua através dos túneis estabelecidos até às redes remotas. Face às suas características, a utilização do TIB/Rendezvous é mais indicada para interligar redes locais ou sítios bem identificados, do que para uso generalizado na *Internet*.

2.5 Sumário

Neste capítulo apresentámos o paradigma de publicação e subscrição de informação. Neste paradigma editores produzem notificações de eventos para o sistema, que as entrega aos subscritores de acordo com os interesses manifestados por estes. Comparado com a comunicação cliente/servidor, este paradigma permite a troca de mensagens em simultâneo entre vários processos, de forma desligada e anónima. Deste modo os sistemas ficam mais flexíveis, facilitando a sua manutenção e evolução.

A forma de subscrição da informação e a riqueza dos mecanismos de selecção, dependem do método de endereçamento utilizado. Assim, apresentámos os endereçamentos *por-canal*, *por-assunto*, e *por-conteúdo*. Este último é o mais versátil dos três pois permite seleccionar as notificações de acordo com os atributos destas. No entanto, nenhum destes métodos de endereçamento permite a definição de zonas delimitadas de difusão de notificações.

Apesar da variedade de soluções que realizam este paradigma de comunicação, podemos identificar dois modelos principais: o de rede de nós distribuidores e o de difusão. No primeiro caso a capacidade de retenção de conhecimento das subscrições em cada nó possibilita a utilização de mecanismos de filtragem ricos, pelo que estas soluções oferecem em geral o endereçamento *por-conteúdo*. No entanto, a entrega de notificações aos subscritores finais é efectuada através de ligações ponto-a-ponto, sendo menos eficiente que por difusão. No modelo de difusão, a entrega é efectuada por protocolos de comunicação em grupo ao nível da rede, seja por difusão de um único pacote para todos os subscritores (*broadcast*) ou para um subconjunto destes (*multicast*). Contudo, este modelo dificulta a realização de métodos de endereçamento mais elaborados, sendo normalmente suportado apenas o endereçamento *por-assunto*. Convém referir neste modelo a necessidade de emparelhar expressões de subscrição em endereços de grupo. Esta operação sendo trivial no caso da difusão total, não o é na difusão por grupos, tal como veremos mais adiante no Capítulo 4.

A maioria das soluções apresentadas distribui a informação através de uma rede de ligações ponto-a-ponto entre nós servidores de eventos (Elvin[8], SIENA[9], Gryphon [24]). Outras não utilizam qualquer servidor intermédio entre editores e subscritores de informação, e recorrem a mecanismos de difusão total em *LANs* (TIB/Rendezvous[6]). Vimos também as razões dessas soluções apresentarem problemas de escala quando aplicadas na *Internet*, no primeiro caso pela necessidade de difundir por toda a rede os anúncios dos assuntos, e no segundo caso porque a difusão total só pode ser praticada em redes locais controladas. Em geral, estas soluções ignoram os mecanismos de difusão da rede baseados no *IP Multicast*, nomeadamente como veículo de entrega de notificações.

Capítulo 3

Técnicas para maior escala

Em geral, as soluções baseadas nos modelos anteriores só são aplicáveis em redes locais, devido aos constrangimentos ou problemas que o seu desenho apresenta quando inseridas em redes de maior escala como a *Internet*. As que têm tido aplicações em redes mais alargadas fazem-no em ambientes controlados, normalmente através de conexões ponto-a-ponto em forma de túnel entre ilhas de publicação na rede.[1] Como tal, estas soluções não oferecem uma base para um serviço de publicação e subscrição de informação de grande escala.

É objectivo deste capítulo propor um conjunto de técnicas, que possam servir de base à criação de um serviço generalizado de publicação e subscrição de informação, com capacidade de escala em redes alargadas (*WAN*). As técnicas propostas derivam de técnicas fundamentais dos dois modelos de publicação apresentados, como sejam, a constituição de uma rede de nós distribuidores de eventos e o uso de emparelhamentos de expressões de subscrição em endereços de grupo *IP Multicast*. As contribuições introduzidas dizem respeito à constituição da rede de nós em domínios de publicação, à subscrição orientada aos assuntos de cada domínio, ao suporte bastante eficiente dos

[1] O TIB/Rendezvous é um bom exemplo disso. É também um dos sistemas mais utilizados.

métodos de endereçamento por-assunto e por-conteúdo, ao algoritmo de emparelhamento de subscrições em endereços de grupo *IP Multicast*, e à faculdade de se poder efectuar controlo de acesso aos assuntos de cada domínio.

3.1 Domínios de publicação

Nesta tese propomos um modelo de espaço de informação concretizado por uma topologia de rede de nós assente na criação de *domínios de publicação* na *Internet*.

Ao nível da infra-estrutura um domínio é formado por um ou vários segmentos de rede de tal modo que exista sempre um caminho interior a ligar quaisquer duas máquinas dentro do domínio. Um domínio equivale a uma *zona administrativa* (do inglês, *administrative scope*) configurada recorrendo ao protocolo *MZAP* (*Multicast-Scope Zone Announcement Protocol*) [28], um dos mecanismos previstos no contexto da *Internet Multicast Address Allocation Architecture* [27] (Apêndice A). Segundo esta arquitectura de difusão na *Internet*, uma zona administrativa consiste numa região da rede na qual está defina uma gama de endereços de grupo *IP Multicast* (*multicast scope*), tal que um pacote emitido para um desses grupos é confinado à fronteira topológica da região. Este mecanismo de controlo é realizado pelos encaminhadores de difusão (*multicast routers*) na fronteira do domínio, que ao receberem um pacote emitido para um dos grupos da gama não o deixam passar para o exterior da região. Assim, uma zona administrativa facilita a gestão do domínio (em particular a configuração da sua fronteira topológica), suplantando técnicas menos eficazes de circunscrição de tráfego de difusão, como seja o uso do campo *TTL* dos pacotes *IP*.[2]

De forma a poder ser referenciado, um domínio possui um identificador numérico

[2]Antes da criação do conceito de zona administrativa, as aplicações *IP Multicast* só podiam confinar a emissão de pacotes a uma região através da utilização do campo *TTL* (*Time-To-Live*). Contudo, este método não é totalmente fiável e provoca alguns problemas quando aplicado juntamente com alguns protocolos de difusão na rede RFC 2365 [29].

único (o seu endereço) ao qual poderão estar associados vários nomes de um espaço hierárquico de nomes de domínios, à semelhança do serviço *DNS*[3].

Um domínio pode conter vários sub-domínios ou intersectar com outros domínios vizinhos, sendo obrigatório que cada domínio defina a sua gama de endereços de difusão, de forma a não intersectar com as definidas pelos restantes. No primeiro caso, os nomes dos domínios filhos deverão descender do nome do domínio pai de modo a evidenciar a sua inclusão neste último. Por exemplo, o domínio 'sun.com' poderá conter os domínios 'java.sun.com' e 'hardware.sun.com'.

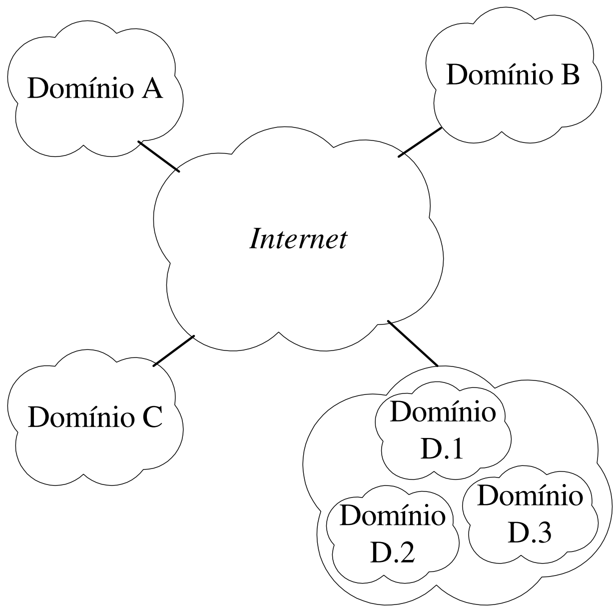


Figura 3.1: Domínios de publicação na *Internet*.

A gestão do espaço de informação do domínio e da sua fronteira topológica cabe à respectiva *autoridade de publicação*. Como veremos, esta possui um papel determinante no controlo de acesso ao domínio.

[3]Propomos a utilização do *DNS* como serviço de nomes de domínios de publicação, e o uso de identificadores *MZAP* de zona administrativa (*ZoneIDs*) como identificadores destes.

3.1.1 Domínios geográficos

Os nomes de domínio podem representar uma região remota da rede ou uma região local. Neste último caso o nome toma um sentido geográfico, transversal a qualquer outro domínio. Estes nomes locais dão suporte ao desenvolvimento de aplicações que usem serviços independentemente da região a que pertencem como é o caso das aplicações móveis. Assim, sugerimos a utilização dos nomes *localhost* para referenciar o domínio da máquina local, e *localdomain* para referenciar o domínio da rede local. De modo a suportar futuras aplicações, todos os nomes com sentido geográfico e prefixo *local*, deverão ser reservados.

3.1.2 Entidades do domínio

Em cada domínio existe um conjunto de entidades membro que realizam o paradigma de publicação e subscrição de informação. Estas entidades executam as três funções nucleares deste paradigma de comunicação, a saber, a publicação, a subscrição e a distribuição de informação. Estas entidades do sistema são normalmente conhecidas por,

- ***editores***, publicam informação no sistema via a invocação da primitiva *publicar()*. Simultaneamente, propomos o uso de mensagens de anúncio e a introdução das primitivas *anunciar()* e *cancelarAnúncio()* (§3.5);

- ***subscritores***, invocam interesse na informação publicada através da primitiva *subscrever()*;

- ***servidores***, responsáveis pelo processamento de anúncios de novas publicações, pelo processamento dos pedidos de subscrição, pelo controlo de acesso à informação (§3.7), pela execução do algoritmo de emparelhamento de subscrições em endereços de grupo *IP Multicast* (§4), e pela entrega das notificações aos subscritores.

Simultaneamente, introduzimos três papéis distintos e acumuláveis para definição das responsabilidades de qualquer servidor de eventos participante no sistema, a saber:

- ***servidor fronteiriço***, garante o controlo de acesso aos assuntos públicos do domínio, e verifica se os pedidos de subscrição reúnem os privilégios necessários. Este servidor toma conhecimento dos domínios a que pertence por captura das mensagens *ZAM* (*Zone Announcement Message*) inerentes à execução do protocolo *MZAP*. Os nomes dos domínios coincidem com os nomes das respectivas zonas definidas via *MZAP*;

- ***servidor de acesso***, é a ponte de ligação dos editores e dos subscritores com o sistema. Também aprende pelas mensagens *ZAM* os domínios a que pertence;

- ***servidor distribuidor***, realiza o encaminhamento dos pedidos de subscrição e das notificações, no primeiro caso no sentido dos domínios a que respeitam os assuntos, no segundo caso no sentido dos subscritores à escuta. Este tipo de servidor pode não pertencer a qualquer domínio, fazendo então parte da rede entre domínios.

A Figura 3.2 ilustra cada uma das entidades descritas e respectivos papéis dentro do sistema de publicação e subscrição. Nesta figura os domínios são delimitados por elipses. Os servidores de eventos estão ligados entre si, e as letras identificam-nos como de (F)ronteira, de (A)cesso, ou simplesmente (D)istribuidores. Note-se que qualquer servidor tem sempre o papel de distribuidor.

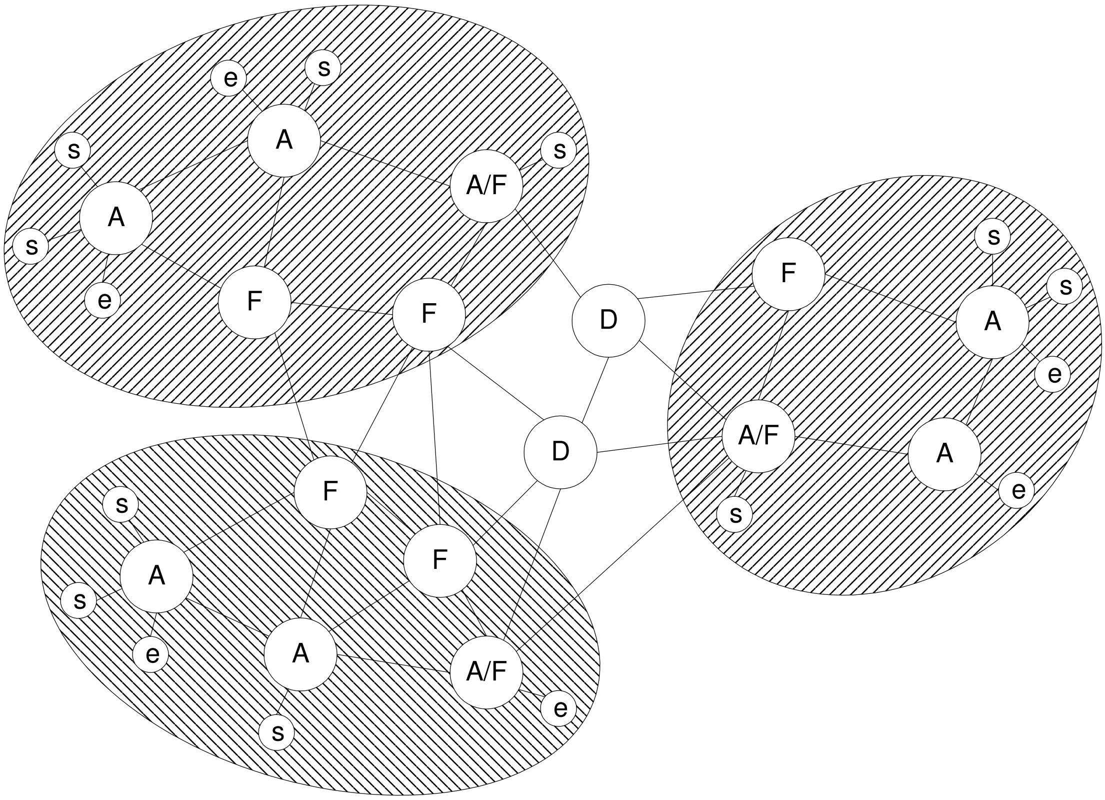


Figura 3.2: Entidades dos domínios de publicação.

3.2 Espaço de informação

A introdução de domínios de publicação suporta a criação de um modelo de espaço de informação de $(1 + n)$-dimensões por cada domínio. Este modelo de espaço de informação é uma variante do método de endereçamento por-conteúdo apresentado (§2.1.3), na qual se entende este método de endereçamento como uma extensão do método endereçamento por-assunto, constituindo-se o nome de assunto como o único ponto comum entre os dois métodos de endereçamento. Assim, o espaço de informação de um domínio é formado por uma dimensão obrigatória, o nome do assunto dentro do domínio, a qual determina as restantes $n \geq 0$ dimensões do sub-espaço de informação desse assunto. Esta concepção do endereçamento por-conteúdo possibilita a definição de espaços de informação tipificados de acordo com a natureza da informação veiculada em cada assunto, isto é, as dimensões diferem entre assuntos e representam propriedades da informação neles publicada.

Assim, se $n > 0$, o assunto é representado por uma classe de eventos de estrutura conhecida tanto pelos seus editores como pelos seus subscritores[4]. Em cada classe, o atributo a_k é definido pelo par ($nome_k$, $tipo_k$). O tipo pode ser discreto, como numérico, lógico, data, cadeia de caracteres e conjunto enumerável, ou contínuo (intervalo limitado em $\Re$). Por exemplo, podemos ter as seguintes definições de atributo: (”idade”, inteiro); (”cor”, {vermelho, azul, amarelo}); e (”peso”, [0; 200]).

Em cada assunto os eventos são expressos pela difusão de notificações, cujo conteúdo possui uma estrutura definida pela classe de eventos respectiva. Assim, o conteúdo de cada notificação é formado pela instanciação de valores para o conjunto dos atributos da respectiva classe. Adicionalmente, a notificação poderá ainda conter outra informação não acessível ao sistema de publicação e subscrição de informação, ilustrada a cinzento na Fig. 3.3.[5]

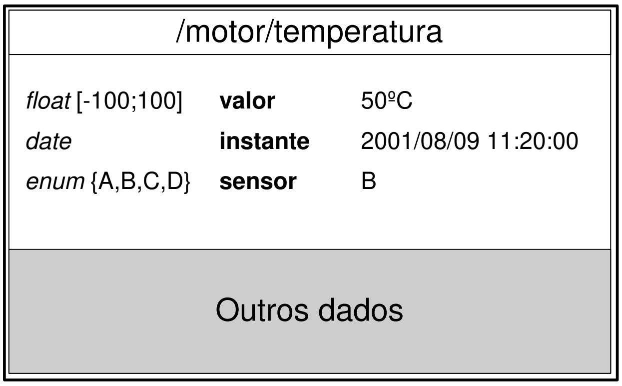


Figura 3.3: Exemplo de uma notificação para o assunto ”/motor/temperatura”.

Esta interpretação do endereçamento por-conteúdo facilita a adição de novos assun-

[4]Isto não significa que o sistema de publicação e subscrição de informação tenha de ser orientado por objectos.

[5]Se imaginarmos um sistema de publicação e subscrição de informação orientado por objectos, é possível que uma classe realize interfaces adicionais não acessíveis ao sistema de publicação.

tos ao sistema em qualquer altura, sendo suficiente definir o esquema de propriedades do assunto a acrescentar.[6]

3.2.1 Expressão de endereçamento

Em qualquer sistema de publicação e subscrição deverá ser definido um esquema para referenciar a informação, a usar tanto na publicação como na subscrição desta. A sintaxe utilizada para referenciar a informação deverá ser a mais simples a suportar o modelo de espaço estabelecido. No caso presente, sendo facultados o endereçamento por-assunto e por-conteúdo, o esquema de referências toma o formato de uma *URL* (*Uniform Resource Locator*) [30] [7] tal como segue,

$$\underbrace{\underbrace{<\mathit{proto}>:\ //\ <\mathit{domínio}>\ /\ <\mathit{assunto}>}_{\text{por-assunto}}\ [?\ <\mathit{filtro}>]}_{\text{por-conteúdo}}$$

Figura 3.4: Sintaxe de referência da informação.

Como se observa na figura, o esquema de endereçamento destaca o método por-conteúdo como extensão do por-assunto, por aplicação de um filtro ao fluxo de notificações. Comentando a figura, $<\mathit{proto}>$ identifica o nome do protocolo a atribuir ao esquema proposto. O $<\mathit{domínio}>$ indica o domínio de publicação ao qual pertence o $<\mathit{assunto}>$, assumindo o nome deste último um formato hierárquico, tal como, "/bolsa/acções/XPTO". Pertencendo as notificações emitidas à classe de eventos do $<$*assunto*$>$, o $<$*filtro*$>$ representa uma condição lógica sobre os valores dos atributos da classe, possibilitando a selecção das notificações. No $<$*filtro*$>$ podem ser usados os operadores lógicos usuais, $=$, $\neq$, $<$, $>$, $\leq$ e $\geq$, bem como as ligações lógicas $\vee$ e $\wedge$.

[6]Compare-se com a interpretação de endereçamento por-conteúdo apresentada em §2.1.3, na qual as

$$pub : //lusa.pt/internacional/Kosovo$$
$$pub : //dn.pt/economia/telco$$
$$pub : //desporto.pt/futebol/Sporting$$
$$pub : //bvl.pt/EDP?valor > 5000\$00$$
$$pub : //sun.com/java/jdk?action = "update" \wedge version > 1.4$$
$$pub : //localhost/filesystem?action = "change" \wedge filename = "passwd.txt"$$

Figura 3.5: Exemplos de referências para assuntos.

3.3 Subscrição orientada ao domínio

A introdução de domínios de publicação identificados por nomes permite criar o conceito de *subscrição-orientada* ao domínio.

Como apresentado na Fig. 3.4, o nome do domínio faz parte da sintaxe das expressões de endereçamento da informação independentemente do método utilizado. A introdução de nomes de domínio no acto de endereçamento permite especificar qual é o domínio fonte da informação referenciada. No esquema proposto, o nome do domínio é usado pelos editores para indicar qual a região de propagação da informação, e pelos subscritores para indicar de que domínio pretendem receber informação.

Somente os editores contidos num domínio podem anunciar e publicar para o seu espaço de informação. Assim, ao contrário de soluções anteriores nas quais é necessário usar mecanismos de difusão total de anúncios pela rede de distribuidores[8], nesta nova técnica a difusão das mensagens de anúncio é limitada à fronteira do domínio a que respeitam, pelo que se reduz o número deste tipo de mensagens na rede. Por seu lado, as subscrições são encaminhadas directamente para o domínio fonte endereçado,

dimensões do espaço de informação são pré-definidas.

[7]Sendo portanto similar ao mecanismo de endereçamento actualmente usado na *WWW*.

[8]No pior dos casos recorrem à difusão de subscrições por toda a rede, mas nem sequer essas soluções são aqui consideradas em virtude da sua ineficiência em sistemas de grande escala.

e consequentemente, para perto dos editores da informação. Portanto, o conceito de *subscrição-orientada* ao domínio possibilita o aumento da capacidade de escala comparativamente com soluções anteriores, pois não é necessário que cada distribuidor tenha conhecimento da localização das fontes de todos os assuntos (tal como acontece no SIENA [9]).

Simultaneamente, num serviço de publicação e subscrição genérico e aberto, particularmente se realizado na *Internet*, a subscrição orientada pode revelar-se bastante importante por motivos de segurança, pois eventualmente é do interesse dos consumidores de informação conhecer quem a produz e poder confiar no seu conteúdo.[9]

3.4 Relações de cobertura

Antes da descrição das primitivas do sistema, introduzimos o suporte teórico necessário. As relações de semântica base em sistemas de publicação e subscrição de informação com endereçamento por-conteúdo, dizem respeito ao conceito de cobertura entre notificações e subscrições, e entre subscrições e anúncios se estes últimos forem utilizados [9].

3.4.1 Semântica dos filtros de atributo, $\phi \subset_f^n \alpha$

Denota-se por $\phi \subset_f^n \alpha$ a relação de cobertura entre um filtro de atributo ϕ e o respectivo atributo α duma notificação. Esta relação é verdadeira se e só se a conjunção

$$\phi.nome = \alpha.nome \wedge \phi.tipo = \alpha.tipo \wedge \phi.operador(\alpha.valor, \phi.valor) \tag{3.1}$$

[9]Na introdução a esta dissertação (§1) salientámos a questão de perda do anonimato da fonte de informação. Teoricamente, o anonimato total poderia ser alcançado pela definição de um domínio global. Contudo, atendendo à reduzida capacidade de escala dum domínio destas dimensões tal não será viável no presente.

for verdadeira. Esta expressão, indica que um filtro de atributo cobre um atributo duma notificação, quando ambos têm o mesmo nome e tipo, e o valor do atributo emparelha com o valor do filtro, atendendo ao operador definido por este. Neste caso sendo válida a condição, diz-se que "o filtro de atributo ϕ cobre α" ou que "α emparelha com o filtro de atributo ϕ", ou mais simplesmente, "ϕ cobre α" ou "α emparelha com ϕ". Por exemplo, para a notificação da Fig. 3.3 e para os filtros de atributo $\{float[-100;100]\ valor\ >\ 0^oC\}$ e $\{enum\{A,B,C,D\}\ sensor\ \neq\ A\}$ são válidas as seguintes relações

$$valor.nome = \mathbf{valor}.nome \wedge valor.tipo = \mathbf{valor}.tipo \wedge valor.>(50^oC, 0^oC)$$
$$sensor.nome = \mathbf{sensor}.nome \wedge sensor.tipo = \mathbf{sensor}.tipo \wedge sensor.\neq(B, A)$$

3.4.2 Semântica das subscrições, $s{\subset}^n_s n$

A relação $\subset^n_f$ serve de base à definição da relação de cobertura de uma notificação n por uma subscrição s. Esta relação denota-se por $s{\subset}^n_s n$ e é válida para o conjunto de notificações $N_S(s) = \{n \in N_a : \forall\phi \in s, \exists\alpha \in n : \phi{\subset}^n_f\alpha\}$, representando N_a o conjunto de todas as notificações do assunto a.

Para completar, podemos estabelecer a relação de cobertura entre duas subscrições denotada por $s{\subset}^s_s s'$. Esta relação é dada pela expressão $s{\subset}^s_s s' \Leftrightarrow N_S(s){\supseteq}N_S(s')$, e sendo transitiva tem-se sempre $\forall s, s', s'' : s{\subset}^s_s s' \wedge s'{\subset}^s_s s'' \Rightarrow s{\subset}^s_s s''$.

Como veremos, as relações de cobertura $\subset^n_s$ e $\subset^s_s$ realizam a semântica do encaminhamento de notificações e de pedidos de subscrição, respectivamente.

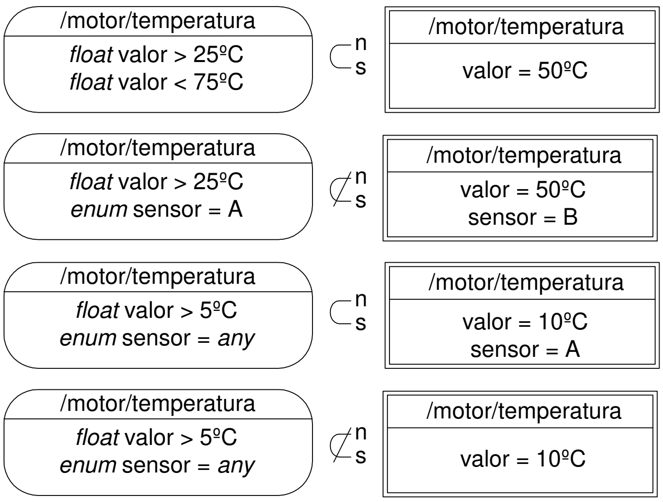


Figura 3.6: Exemplo da relação de cobertura $s \subset_s^n n$.

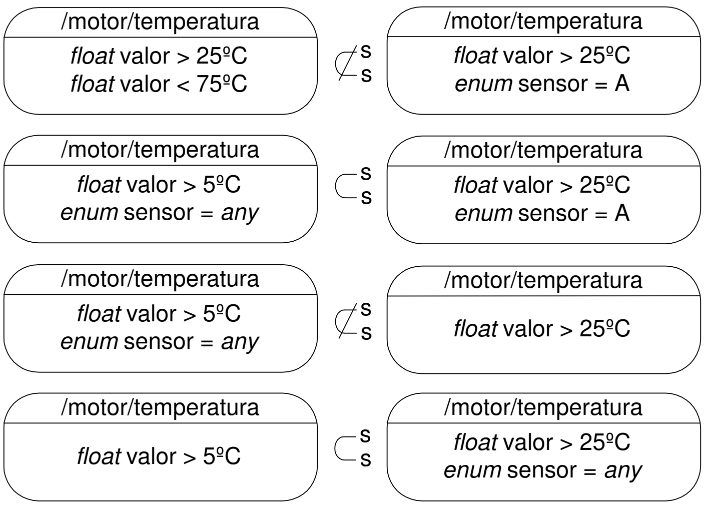


Figura 3.7: Exemplo da relação de cobertura $s \subset_s^s s'$.

3.4.3 Semântica dos anúncios, $a \subset_a^s s$

Num sistema que dê suporte ao mecanismo de anúncio, é necessário definir a semântica na base do seu funcionamento. Esta é denotada pela relação $a \subset_a^s s$, a qual estabelece a relação de cobertura entre uma subscrição s e um anúncio a para um mesmo assunto. A relação $a \subset_a^s s$ é válida se e só se $N_A(a) \cap N_S(s) \neq \emptyset$, onde $N_A(a) = \{n \in N_a : \forall \alpha \in n, \exists \phi \in a : \phi \subset_f^n \alpha\}$. Tal como definido $N_A(a)$ representa o conjunto de todas as notificações tais que, para qualquer atributo α duma notificação, existe definido no anúncio a um filtro de atributo ϕ, para os quais é verdadeira a relação $\phi \subset_f^n \alpha$. Diz-se então que o anúncio a é relevante para a subscrição s, ou que a subscrição s é compatível com o anúncio a.

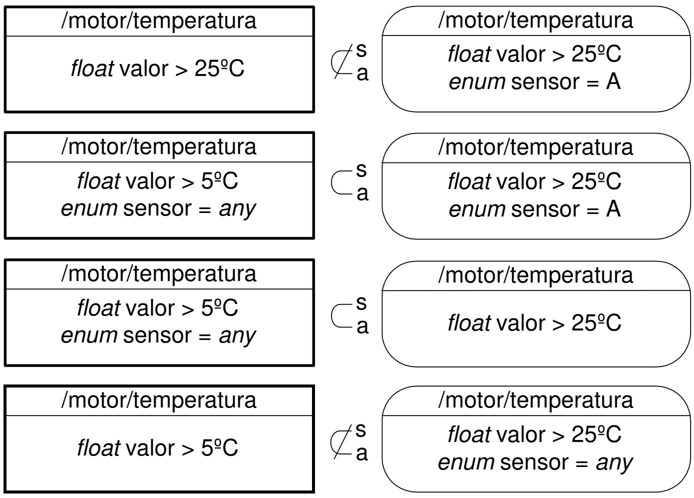


Figura 3.8: Exemplo da relação de cobertura $a \subset_a^s s$.

Por fim, define-se a relação $a \subset_a^a a'$ válida para $N_A(a) \supseteq N_A(a')$. Tal como será dado a conhecer mais adiante, as relações $\subset_a^s$ e $\subset_a^a$ revelam-se bastante importantes na concretização dos mecanismos de encaminhamento de subscrições e de anúncios, respectivamente.

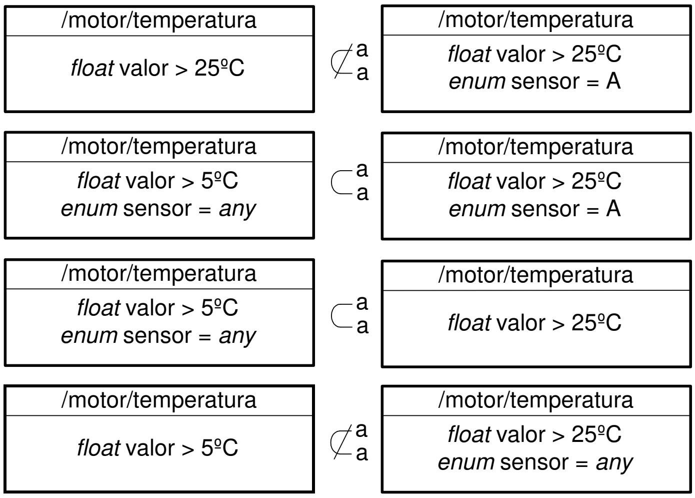


Figura 3.9: Exemplo da relação de cobertura $a \subset^a_a a'$.

3.5 Primitivas

Nesta secção acrescentamos relativamente a (§2.2) duas outras primitivas, *anunciar()* e *cancelarAnúncio()*, e apresentamos as funcionalidades básicas do modelo de publicação e subscrição proposto neste trabalho. O quadro seguinte lista todas as primitivas definidas.

anunciar()	comunicar a publicação de um assunto
cancelarAnúncio()	cancelar o fim da publicação de um assunto
publicar()	enviar uma notificação sobre um assunto
subscrever()	comunicar interesse em toda ou numa parte da informação de um assunto
cancelarSubscrição()	comunicar o desinteresse por informação anteriormente subscrita

Figura 3.10: Primitivas do sistema.

A Fig. 3.11 demonstra a relação entre cada uma destas primitivas e o esquema de endereçamento proposto anteriormente (§3.2.1). Ao contrário das outras funções, na publicação de uma notificação não é necessário indicar um filtro. O uso de um filtro num anúncio permite indicar a publicação de um sub-espaço de informação dum assunto. Analogamente, um filtro numa subscrição endereça um sub-espaço sobre o qual se tem interesse em receber notificações. No caso das operações de cancelamento, o filtro indica o sub-espaço para o qual se deixou de publicar ou do qual se deixou de ter interesse, respectivamente para *cancelarAnúncio()* e *cancelarSubscrição()*.

$$\underbrace{\underbrace{<\textit{proto}>\text{: // }<\textit{domínio}>\text{ / }<\textit{assunto}>}_{\textit{publicar()}}\ \text{[? }<\textit{filtro}>\text{]}}_{\textit{anunciar(), cancelarAnúncio(), subscrever(), cancelarSubscrição()}}$$

Figura 3.11: Sintaxe de endereçamento da informação vs. primitivas do sistema.

3.5.1 Anúncio

Dentro de um domínio, qualquer editor avisa que assuntos vai publicar através da primitiva *anunciar()*, a qual resulta no envio de uma mensagem para o servidor de acesso local. Este último, só aceita o anúncio se o domínio endereçado corresponder a um dos domínios do qual é membro, pois somente os editores dentro do domínio é que podem anunciar e publicar para assuntos deste. Qualquer servidor que receba uma mensagem de anúncio, encaminha-a pela rede de servidores do domínio, se e só se, não tiver anteriormente enviado algum anúncio que cubra o recebido, de acordo com a relação $\subset_a^a$. Assim, a relação $\subset_a^a$ revela-se útil na redução do número de mensagens de anúncio na rede, promovendo uma maior capacidade de escala.

3.5.2 Subscrição

O conceito de *subscrição-orientada* pressupõe o encaminhamento dos pedidos de subscrição até aos domínios endereçados. Deste modo, quando um subscritor submete um pedido de subscrição ao seu servidor de acesso, este deverá obter o identificador do domínio a partir do nome presente na expressão de endereçamento. Uma vez identificado o domínio, o servidor de acesso tem de determinar a qual dos servidores de eventos seus vizinhos deve encaminhar o pedido de subscrição. Este é escolhido em função do domínio de destino e de acordo com uma tabela de encaminhamento.

Na sua forma mais simples, esta tabela regista para cada domínio destinatário, o servidor de eventos para o qual devem ser enviadas as subscrições de assuntos desse domínio. Note-se que todos os servidores têm a sua tabela de encaminhamento, independentemente do seu papel. No entanto, cabe somente aos servidores fronteira comunicar a identidade dos domínios a que pertencem[10], aos servidores vizinhos fora desses domínios. Neste caso, é reportada uma distância nula. Entretanto, qualquer servidor de eventos reporta a distância a qualquer domínio ao qual não pertença[11]. No seu conjunto, a informação contida nestas tabelas regista o caminho mais curto para qualquer domínio destino/fonte. Atendendo à natureza e semelhança destas tabelas com as usadas para encaminhamento *IP*, poderão ser adaptadas técnicas já desenvolvidas para este último protocolo.

À semelhança do que acontece com as mensagens de anúncio, qualquer servidor que receba um pedido de subscrição, só o entrega ao servidor seguinte se e só se, não tiver anteriormente entregue uma subscrição que cubra a actual, atendendo à relação $\subset_s^s$. Mais uma vez, se revela a utilidade das relações de semântica apresentadas anteriormente, na promoção de ganhos na capacidade de escala do sistema de eventos.

Por fim, quando os pedidos de subscrição chegam junto de um servidor de fronteira do domínio fonte, a informação das mensagens de anúncio permite optimizar o encaminhamento dos pedidos internamente, dirigido-os pelo caminho mais curto, no sentido

[10]Relembra-se que qualquer servidor sabe a que domínios pertence através do protocolo *MZAP*.

[11]A distância a um servidor vizinho deverá medir-se pelo custo de entrega de uma mensagem a esse servidor. O custo será uma função de uma ou mais variáveis características da ligação a esse servidor.

dos servidores de acesso junto dos editores que fizeram anúncios compatíveis.

3.5.3 Publicação

A publicação é efectuada pela invocação da primitiva $publicar()$, sendo-lhe passados como argumentos uma referência para o assunto e a notificação a emitir (atributos e outros dados extra). Cabe ao servidor de acesso verificar se a notificação diz respeito a um assunto do domínio, anunciado pelo editor em questão. Somente neste caso, é que a notificação é encaminhada pela rede de servidores de eventos, pelo caminho em árvore mais curto (*minimal spanning tree*) até aos subscritores, estabelecido pelas subscrições anteriormente efectuadas por estes.

Contudo, um editor só entrega uma notificação n ao seu servidor de acesso, se e só se, este o tiver informado da existência de subscritores interessados, ou seja, somente se o editor tiver recebido do seu servidor de acesso uma expressão de subscrição s tal que se verifique $s \subset_s^n n$. Esta capacidade para saber previamente da existência de subscritores, evita sobrecarregar desnecessariamente o sistema com notificações não pretendidas.[12] A expressão de subscrição s corresponde à reunião de todas as subscrições recebidas pelo servidor de acesso para um mesmo assunto. Cabe a este último, avisar todos os editores que submeteram anúncios a tal que $a \subset_a^s s$, sempre que houver uma alteração de s. Este aviso poderá ser feito ponto-a-ponto ou por difusão para um endereço de grupo conhecido.

3.5.4 Listagem de assuntos

Um subscritor poderá ter acesso à lista de assuntos públicos dum domínio, bastando enviar para este um pedido de listagem através da rede de servidores de eventos. Este

[12] O Elvin[8] possui um mecanismo semelhante para redução de tráfego na rede, ao qual atribui o nome de *"quenching"*.

pedido é interceptado por qualquer um dos servidores na fronteira do domínio, o qual enviará a resposta de volta ao subscritor. No nome de assunto, poderão ser usadas expressões regulares para filtragem dos assuntos do domínio endereçado.

Outro mecanismo, eventualmente mais simples, poderá passar pela utilização de um servidor *web* por domínio.

3.6 Entrega final das notificações

Por comparação com anteriores sistemas que concretizam o modelo de rede de nós (Elvin[8], SIENA[9] e Usenet News[17]), onde a entrega final das notificações aos subscritores é realizada via ligações ponto-a-ponto, os servidores de acesso propostos utilizam a difusão ao nível da rede sempre que isso se justifique, recorrendo aos mecanismos de endereçamento de grupos em *IP Multicast*.

Uma vez que cada servidor de acesso reúne os interesses dos seus subscritores locais, é possível agrupar as subscrições efectuadas, tomando em consideração a semelhança dos respectivos sub-espaços de informação. Pretende-se desta forma reduzir o consumo de largura de banda resultante do envio repetido de uma mesma notificação para um conjunto de subscritores. Como tal os *servidores de acesso* empregam um algoritmo de emparelhamento de subscrições em endereços de grupo locais ao domínio[13].

3.7 Controlo de acesso

Os conceitos de *domínio de publicação* e de *subscrição-orientada* ao domínio, permitem introduzir na arquitectura mecanismos de controlo de acesso à informação, uma vez que

[13] Pertencentes à *multicast scope* definida para o domínio a que pertencem ou, se este for grande de mais, à *localscope* em que se inserem. Eventualmente poderá ser necessário utilizar um protocolo da reserva de endereços de grupo, tal como o *MADCAP* (do inglês, *Multicast Address Dynamic Client Allocation Protocol*) [31] caso existam outras aplicações em disputa por endereços de grupo.

a difusão desta pode ser limitada à fronteira do domínio. Consequentemente, a informação publicada em cada assunto pode ser pública ou privada, cabendo à autoridade de publicação no domínio definir que assuntos podem ser exportados.

Estas configurações de acesso são comunicadas aos servidores de fronteira, através do envio de uma mensagem para o grupo local de gestão do domínio[14]. Assim, um servidor fronteiriço só aceita um pedido de subscrição, se e só se, corresponder a um assunto público do domínio ou reunir as credenciais exigidas.

Como se poderá facilmente verificar, este tipo de controlo é muito difícil de realizar em sistemas em que os editores podem produzir para qualquer assunto independentemente da sua localização na rede, ou seja, em sistemas onde o direito de publicação não pode ser controlado. Como tal, estes sistemas não fornecem a plataforma necessária para a realização do paradigma de publicação e subscrição de informação, de modo aberto e generalizado, na *Internet*.

Por fim, caso seja necessário assegurar a identidade da fonte e a integridade da informação emitida, as notificações podem ser assinadas pelo domínio.

3.8 Tamanho das mensagens

Um outro aspecto importante diz respeito ao tamanho das mensagens usadas, quer sejam de notificação ou de execução dos mecanismos do sistema.

O facto das notificações serem entregues recorrendo a grupos de difusão ao nível da rede, influencia esta decisão. Assim, cada evento reportado é entregue num único pacote e como tal não pode exceder o espaço que este reserva para dados. Esta escolha é fundamentada também no facto de se entender uma notificação como um conjunto de dados reduzido, mas suficiente para caracterizar o acontecimento que reporta. Caso seja

[14]Deverá ser usado um *grupo relativo* pertencente à gama de endereços definida via *MZAP*. Ver Apêndice A.

preciso, é sempre possível incluir uma referência para informação mais completa, não se limitando assim o desenvolvimento das aplicações. Por outro lado, esta opção evita a necessidade de reordenar pacotes (prescindindo do uso de *buffers* para este efeito), melhorando os tempos de entrega das notificações. Pela mesma razão, qualquer outro tipo de mensagem do sistema é sempre entregue num único pacote.

3.9 Instalação

Atendendo ao desenho da *Internet Multicast Address Allocation Architecture* [27] e à definição de *domínio de publicação*, a fronteira de um domínio coincidirá com as fronteiras das áreas de um sistema autónomo (*autonomous system, AS*). Como tal, os nós servidores de fronteira deverão ficar perto dos *ASBRs* (*AS Boundary Routers*) e dos *ABRs* (*Area Border Routers*), assegurando desta forma o processamento mais rápido de qualquer pedido vindo do exterior, logo que entre na área do domínio.

Adicionalmente, as conexões entre os servidores de eventos deverão replicar o mais fielmente possível as ligações ao nível da rede entre os encaminhadores, de modo a minimizar o caminho físico percorrido pelas mensagens.

Idealmente, o código de encaminhamento deveria constituir uma camada nos actuais encaminhadores (*routers*) da *Internet*, constituindo-se estes como encaminhadores aplicacionais. A capacidade de escala dum dispositivo com esta funcionalidade poderá ser actualmente questionável, mas atendendo ao progresso rápido do equipamento, de futuro poderá ser uma opção a considerar.

Convém salientar que tal como está desenhado, o mecanismo de encaminhamento possibilita que a instalação de servidores possa ser realizada gradualmente à medida que aumenta a dimensão do sistema. A instalação de uma rede de servidores de eventos na *Internet*, recorrendo às técnicas descritas neste trabalho, assume um padrão se-

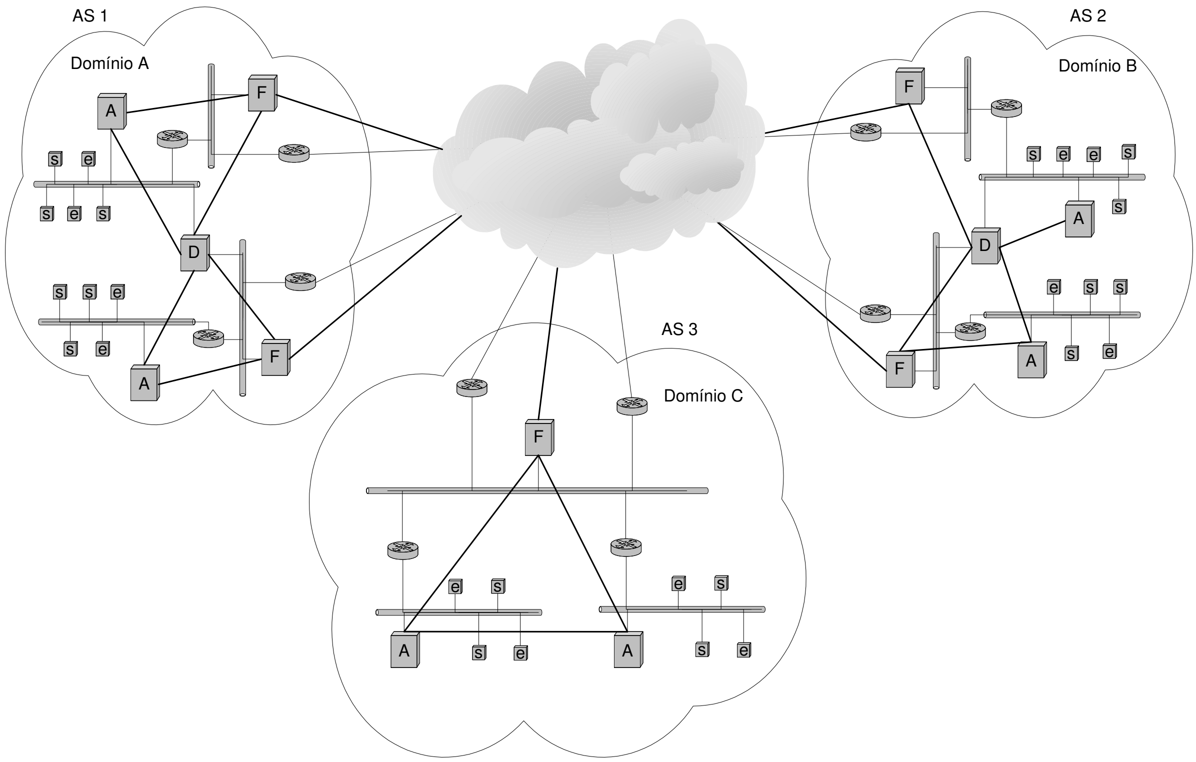


Figura 3.12: Conexões lógicas *versus* conexões físicas ao nível de rede.

melhante ao da instalação da MBONE [32], no qual se procedeu à interligação de ilhas de difusão isoladas usando túneis *IP*. Neste caso, recorre-se simplesmente à interligação lógica de servidores entre domínios.

3.10 Limitações

O dimensionamento de um domínio deverá considerar a quantidade de assuntos a publicar, sob pena de se poder reduzir a capacidade de escala em virtude da necessidade de dispersar os anúncios no seu interior.

A concretização, num protótipo, do conjunto de técnicas propostas neste trabalho, poderá revelar outras limitações neste momento não previstas. Será deixado como trabalho futuro.

3.11 Sumário

No sentido de ultrapassar os problemas das soluções anteriores e de desenvolver uma solução de publicação e subscrição de informação com capacidade de escala na *Internet*, sugerimos a adaptação deste paradigma de comunicação e a utilização dos mecanismos de difusão *IP Multicast*. Em primeiro lugar introduzimos o conceito de *domínio de publicação* como suporte do modelo de espaço de informação proposto. Este espaço de informação centrado em domínios fonte, implica a necessidade de um mecanismo de *subscrição-orientada*, segundo o qual o domínio é endereçado directamente no acto de subscrição. Assim é desnecessária a dispersão de anúncios de assuntos pela rede, proporcionando um aumento da capacidade de escala pela redução de tráfego.

Entendemos que a adaptação do paradigma de publicação e subscrição de informação recorrendo às técnicas apresentadas, não reduz a potencialidade dos sistemas distribuídos desenvolvidos sobre esta forma de comunicação. Uma eventual limitação a existir, diz respeito ao facto dos endereços de subscrição fazerem referência ao nome do domínio fonte. Não obstante, o conceito de *subscrição-orientada* e a atribuição do direito de publicação num domínio somente aos editores nele contidos, trazem vantagens significativas, permitindo aumentar a capacidade de escala deste paradigma, condição necessária para que possa ser genericamente aplicado numa rede de grandes dimensões como a *Internet*.

Capítulo 4

Algoritmo de emparelhamento

A realização de sistemas de publicação e subscrição de informação usando *IP Multicast* para difusão de notificações, revela-se não só como um desafio bastante interessante, mas principalmente como uma solução com grande potencial de aplicação prática. Contudo, do ponto de vista tecnológico são várias as dificuldades incorridas pela escolha do *IP Multicast*. Em primeiro lugar, a reduzida expressividade do mecanismo de endereçamento em grupo dificulta a selecção das notificações de cada assunto e não permite traduzir facilmente as expressões de filtro do endereçamento por-conteúdo. Em segundo lugar, existe o risco de sobrecarga das tabelas de encaminhamento do tráfego *IP Multicast* e de consequente degradação do sistema, em resultado da utilização excessiva de grupos. Este risco apela à necessidade de uma gestão cuidadosa na atribuição de expressões de conteúdo a grupos. Contudo e apesar dos desafios a vencer, existem vantagens na utilização do *IP Multicast*, em particular a possibilidade de envio de um único pacote para qualquer grupo de subscritores. Convém também referir que as capacidades de escala e de evolução desta tecnologia de difusão [32], possibilitam que os sistemas nela assentes possam de forma transparente evoluir e retirar benefícios de novos avanços tecnológicos que eventualmente surjam. Para além disto, é bastante mais fácil e rápido realizar sistemas de publicação e subscrição de informação assentes numa tecnologia comprovada e bem conhecida, tal como é o *IP Multicast*. O algoritmo a seguir apresentado, pretende ultrapassar os problemas anteriores e fazer uso eficiente da tecnologia *IP Multicast* na tradução de endereços por-conteúdo em endereços de grupo.

4.1 Medida de qualidade

Na realização de sistemas de publicação e subscrição de informação com endereçamento por-conteúdo sobre *IP Multicast*, é fundamental desenvolver algoritmos que executem eficientemente o emparelhamento de expressões de subscrição em endereços de grupo. A qualidade destes algoritmos deve ser medida em vários aspectos tais como, o número de grupos usados, o custo das notificações indesejadas entregues aos subscritores por força da reutilização de grupos, e o peso de execução do algoritmo. O algoritmo que desenvolvemos seguidamente toma em consideração cada um destes factores.

4.2 Fundamentos do algoritmo

Na sequência da secção §3.6, o algoritmo de emparelhamento adopta uma estratégia de distribuição local de notificações, correspondentes às subscrições efectuadas pelos subscritores ligados ao servidor de acesso. Neste algoritmo os grupos de distribuição *IP Multicast* são entendidos como canais indiferenciados de difusão de informação, independentemente do conteúdo, origem, e padrão de subscrição. O algoritmo é executado sempre que ocorre um pedido de subscrição, de cancelamento, ou quando se altera o número de grupos de difusão locais. Em qualquer destas situações, a selecção do emparelhamento final é realizada de forma a garantir a qualidade do algoritmo. É também assumida a existência de um conjunto limitado de grupos locais disponíveis para distribuição de notificações. Este limite é localmente configurável pela autoridade do domínio, e serve para eliminar o risco de sobrecarga das tabelas de encaminhamento do tráfego *IP Multicast* local.

Como consequência da imposição deste limite é necessário estabelecer critérios de associação de expressões de subscrição a grupos de difusão. Assim, o algoritmo de emparelhamento é representado matematicamente pela relação $R \subset \{(s, g) : s \in S \wedge g \in G\}$, de atribuição judiciosa de expressões de subscrição a grupos de difusão, onde S constitui o conjunto de todas as expressões de subscrição submetidas localmente pelos subscritores, e G representa o conjunto de todos os grupos locais atribuíveis. Por exemplo, a seguinte figura ilustra um possível conjunto de subscrições,

Subscrições	**Subscritores**
a	A B C
b	A
c	B G
d	A
e	D
g	C
h	C
i	C D
t	E
u	F
v	B G
x	F
z	E

Tabela 4.1: Lista hipotética de subscrições.

As letras maiúsculas representam conjuntos de subscritores (máquinas ou processos) e cada letra minúscula representa uma expressão de subscrição (independentemente do modo de endereçamento ser por-assunto ou por-conteúdo). Se o número de grupos de difusão e a capacidade das tabelas de encaminhamento fossem ilimitados, a tabela anterior constituiria o emparelhamento trivial, onde a cada subscrição caberia a atribuição de um grupo distinto, ao qual se juntavam os respectivos subscritores. Uma vez que essas condições não se verificam na realidade, existe a necessidade de reorganizar o esquema anterior, de modo a responder não só aos interesses de cada subscritor mas também à limitação de recursos. Assim, admitindo um número máximo de sete grupos de distribuição disponíveis, o anterior emparelhamento pode ser reorganizado, por exemplo, nas hipóteses (a) e (b) abaixo, cabendo a cada linha um grupo diferente.

a	A B G
b d	A
c v	B G
g h	C
e i	C D
u x	F
t z	E

(a)

a c v	B G
a b d	A
g h i	C
e i	D
u x	F
t z	E

(b)

Tabela 4.2: Algumas hipóteses de resolução.

Como se pode observar, a partir de uma situação de emparelhamento inicial onde seriam precisos treze grupos de difusão, a reorganização (a) exige apenas sete e a (b) apenas seis, o que se traduz em ambos os casos numa compressão de aproximadamente 50%. Contudo, este exemplo ilustra os dois problemas que podem surgir na determinação de uma solução de emparelhamento, a saber, a recepção de notificações não desejadas e o envio repetido de notificações. Na situação (a), C (quinta linha) recebe notificações relacionadas com a subscrição 'e' que não efectuou, e em (b), as notificações correspondentes a 'a' e 'i' têm de ser enviadas mais do que uma vez para dois grupos de difusão distintos. De um modo geral, qualquer solução de emparelhamento terá sintomas destes dois problemas.

4.3 Processamento de subscrições

Qualquer subscrição s_a é entendida como um par $(a(f), m)$, sendo sempre verdadeira a condição $\forall a{\in}A, \forall m{\in}M\colon\ \forall s_{a,m}, s'_{a,m}\colon\ s_{a,m} \Leftrightarrow s'_{a,m}$, onde $s_{a,m}$ e $s'_{a,m}$ representam subscrições do assunto a pelo subscritor m, e f é um filtro de a. Esta condição indica que qualquer subscritor só pode ter uma única subscrição activa por assunto em cada momento, pelo que essa subscrição é sempre a última que foi aceite em substituição das anteriores. Sempre que é efectuado um pedido de subscrição, é feito o pré-processamento da respectiva expressão de endereçamento por-conteúdo. O filtro de subscrição é simplificado pela redução de redundância, e interpretado como uma sequência de disjunções entre conjunções de restrições lógicas $l_{k,i}(a_i)$ sobre os atributos da classe, ou seja,

$$filtro(s_a) = (l_{1,1}(a_1) \wedge \ldots \wedge l_{1,n}(a_n)) \vee \ldots \vee (l_{k,1}(a_1) \wedge \ldots \wedge l_{k,n}(a_n)) \tag{4.1}$$

Por conseguinte, uma vez que para quaisquer subscrições $s_{a,m} = (a(f' \vee f''), m)$, $s'_{a,m} = (a(f'), m)$ e $s''_{a,m} = (a(f''), m)$, se tem $s_{a,m} = s'_{a,m} \vee s''_{a,m}$, qualquer subscrição s_a é subdividida internamente pelo algoritmo em k subscrições s_{a_k} disjuntas. Deste modo a possibilidade de ocorrência de relações de cobertura entre diferentes subscrições é maior, pelo que o cálculo de emparelhamentos entre notificações e subscrições pode ser mais rápido e envolver menos comparações. Simultaneamente, é possível considerar individualmente cada uma das k subscrições no cômputo de agregações em grupos de difusão.

Uma vez que para qualquer par de assuntos a e b se tem sempre $s_a \not\subset^s_s s_b$, a relação $\subset^s_s$ define-se somente entre subscrições dum mesmo assunto, sendo representada por um grafo das coberturas entre essas subscrições tal como mostra a figura seguinte.

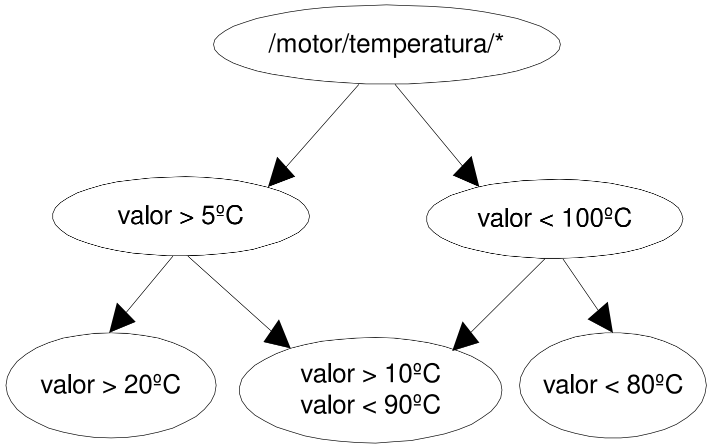


Figura 4.1: Grafo de coberturas $\subset_s^s$ entre subscrições de "/motor/temperatura".

No grafo todos os nós representam subscrições efectuadas, com a eventual excepção do nó raiz que pode corresponder a uma expressão para a qual não foi submetida nenhuma subscrição equivalente. Como os filtros de conteúdo são baseados nos operadores lógicos básicos, esta estrutura em grafo é de manutenção e pesquisa bastante eficientes. No grafo cada nó representa um volume de subscrição e guarda a lista dos subscritores que efectuaram subscrições desse mesmo volume. As setas apontam no sentido dos volumes cobertos, pelo que quem subscreve um determinado volume também subscreve todos os volumes apontados pelas setas que saem do nó correspondente. Portanto, o conjunto dos grafos de cada assunto guarda o estado de todas as subscrições efectuadas localmente, sendo actualizado em função da dinâmica desse estado. Simultaneamente, a reunião da informação de subscrição de cada assunto numa única estrutura independente, possibilita libertar da memória[1] os grafos correspondentes aos assuntos com baixas taxas de produção de notificações, mantendo apenas a informação relativa aos assuntos mais frequentes. Deste modo o sistema de distribuição aumenta o seu desempenho e a sua capacidade de escala por via da adaptação ao padrão de notificações. Como veremos adiante, a identificação das coberturas entre as subscrições dum

[1]Através do armazenando persistente numa unidade local.

único assunto permite rapidamente determinar um conjunto de subscrições candidatas a uma eventual agregação.

4.4 Volume de subscrição

Neste trabalho introduzimos pela primeira vez o conceito de *volume de subscrição* (em inglês, *subscription volume*) $V(s_a)$, como sendo o volume do sub-espaço de notificações do assunto a subscrevido por s_a, e denota-se por $\|V(s_a)\|$ a medida da dimensão desse volume. Considerando a classe dos eventos de a, os atributos a_1, a_2, ..., a_n, constituem o conjunto de eixos do respectivo espaço de n dimensões. A dimensão do volume duma subscrição s_a é calculada em função do seu filtro, $filtro(s_a)$, com base na gama de valores que cada atributo pode tomar no correspondente eixo do espaço multi-dimensional de a, por imposição das restrições aplicadas pelo filtro de s_a. Assim temos que,

$$V(s_a) = V(filtro(s_a)) = V(l(a_1) \wedge \ldots \wedge l(a_n)) \tag{4.2}$$

onde $l(a_i)$ representa o conjunto de valores do eixo i a que o atributo a_i está limitado. Por exemplo, para a classe de eventos {*string* assunto = "/motor/temperatura"}, {*float* valor [-100^{o}C; 100^{o}C]} e para o filtro de subscrição *valor* > 75^{o}C, o volume equivale a todas as notificações de temperaturas superiores a 75^{o}C, ou seja, 12.5% do espaço total de notificações do assunto "/motor/temperatura".

Como consequência da definição de volume de subscrição temos o seguinte,

$$\forall s, s' \in S_a : V(s') \subseteq V(s) \Leftrightarrow s \subset_s^s s' \tag{4.3}$$

4.5 Probabilidade de notificação

O conceito de volume de subscrição é bastante útil, pois não só permite dimensionar o conjunto de notificações subscrevido, como está na base do cálculo da respectiva probabilidade de notificação. Esta define-se como sendo a probabilidade dum subscritor receber notificações após a execução do respectivo pedido de subscrição. Assim, sendo a um assunto qualquer temos,

$$\begin{gathered} P(a) = 1 \\ P(s_a) = \frac{\|V(s_a)\|}{\|V(a)\|} \end{gathered} \tag{4.4}$$

Atendendo ao exemplo anterior vem,

$$P(valor > 75^oC) = \frac{\|valor > 75^oC\|}{\|V(''/motor/temperatura'')\|} = \frac{100 - 75}{100 - (-100)} = 0.125 \tag{4.5}$$

Convém no entanto salientar que tal como definida por (Eq. 4.4), a probabilidade $P(s_a)$ assume uma distribuição uniforme, calculada em função da proporção do volume de subscrição em relação ao volume total do assunto. Não obstante, a distribuição de temperaturas não será certamente uniforme mas sim centrada num determinado valor médio, sofrendo apenas pequenas oscilações à volta deste durante o funcionamento normal do motor. Sendo assim, no caso desta grandeza física o valor obtido pela expressão (Eq. 4.4) serve apenas como estimador inicial, devendo ser substituído por um mais exacto após recolha de uma população de amostras suficiente. O conceito de probabilidade de notificação revela-se útil na estimativa do número de notificações a receber por cada subscrição. Mais adiante, este conceito é usado para prever a possibilidade de entrega de notificações não desejadas aos membros dum grupo de difusão, pelo facto de terem sido agrupados nesse grupo subscrições que esses membros não subscreveram.

4.6 Modelo probabilístico

Qualquer subscrição s_a tem associado o volume $V(s_a)$ do espaço de informação do assunto a, sendo $P(s_a)$ a probabilidade de emparelhamento com uma notificação pertencente a esse volume. Contudo, a função $P()$ tal como definida atrás pressupõe uma distribuição uniforme de eventos. Na realidade, a geração de eventos depende da natureza do sistema supervisionado, pelo que sua distribuição ao longo do tempo pode não ser uniforme em todo o espaço de informação do assunto associado. Por este motivo, é necessário definir a função $P()$ segundo um modelo probabilístico mais rigoroso.

Quando uma máquina distribuidora recebe uma notificação n é feita uma pesquisa a partir da raiz do grafo por todas as subscrições s, tal que $s \subset_s^n n$ seja verdadeira. Para as subscrições encontradas diz-se que ocorreu um emparelhamento (em inglês, *hit*) no nó correspondente, traduzindo-se no incremento de uma unidade ao contador associado ao nó[2]. A informação estatística reunida permite calcular a probabilidade dum novo emparelhamento, como sendo o quociente entre os emparelhamentos ocorridos no nó e o número total de notificações recebidas para o assunto em causa. Se s_a for a subscrição associada, este quociente representa a probabilidade da relação $s_a \subset_s^n n$ ser verdadeira para qualquer notificação que se venha a receber para o assunto a. Consequentemente, o cálculo da probabilidade de notificação duma subscrição deve ser baseado no histórico de emparelhamentos após recolha de uma amostra significativa de notificações, resultando num estimador mais fiável por comparação com o inicialmente definido em (§4.5).

[2]Durante a consulta do grafo um nó pode ser visitado mais do que uma vez. Contudo o incremento é apenas efectuado na primeira passagem.

4.6.1 Função de probabilidade $P()$

A partir dos contadores de cada nó (os seus *hits*) é possível construir um modelo da função de probabilidade $P()$ aplicável a qualquer subscrição. Assim, para uma nova subscrição s_a é determinado um par de subscrições s_a^- e s_a^+ entre as existentes, tais que,

$$s_a \subset_s^s s_a^- \wedge (\forall s{\in}S_a\colon \quad s_a \subset_s^s s \Rightarrow \|V(s_a^-)\| \geq \|V(s)\|) \tag{4.6}$$

$$s_a^+ \subset_s^s s_a \wedge (\forall s{\in}S_a\colon \quad s \subset_s^s s_a \Rightarrow \|V(s_a^+)\| \leq \|V(s)\|) \tag{4.7}$$

Pela definição de s_a^- e de s_a^+ temos sempre o seguinte invariante para a probabilidade de s_a,

$$P(s_a^-) \leq P(s_a) \leq P(s_a^+). \tag{4.8}$$

Sendo a probabilidade uma função dos emparelhamentos efectuados com a subscrição de cada nó, resulta

$$hits(s_a^-) \leq hits(s_a) \leq hits(s_a^+). \tag{4.9}$$

Uma vez que tanto s_a^- e s_a^+ são as subscrições que mais se aproximam de s_a, os valores de $hits(s_a^-)$ e $hits(s_a^+)$ correspondem ao menor intervalo para o valor de $hits(s_a)$. Consequentemente, constituem-se como uma boa base de estimação para o valor de $hits(s_a)$. Assim temos,

$$hits(s_a) = \left\lfloor hits(s_a^+) \cdot \frac{\|V(s_a)\| - \|V(s_a^-)\|}{\|V(s_a^+)\|} + hits(s_a^-) \right\rfloor. \tag{4.10}$$

Eventualmente, durante a determinação de s_a^- e s_a^+ poderão acontecer os seguintes casos:

- se $s_a^- = \emptyset$ (s_a^- não existe): $\quad hits(s_a^-) = 0$;
- se $s_a^+ = \emptyset$ (s_a^+ não existe), pode suceder um dos seguintes casos:

– se o grafo está vazio: s_a é a primeira raiz do grafo e $hits(s_a) = 0$;

– se s_a é uma nova raiz: $hits(s_a) = hits(s_a^{raiz})$;

– se $s_a^{raiz} \not\sqsubset_s^s s_a$: $s_a^{raiz'}$ (menor cobertura de s_a e s_a^{raiz}) é a nova raiz[3] e

$$hits(s_a^{raiz'}) = hits(s_a^{raiz})$$

$$hits(s_a) = \left\lfloor hits(s_a^{raiz'}) \cdot \frac{\|V(s_a)\|}{\|V(s_a^{raiz'})\|} \right\rfloor .$$

Partindo deste estimador, $P(s_a)$ pode ser definido como

$$P(s_a^{raiz}) = 1$$

$$P(s_a) = \begin{cases} \frac{\|V(s_a)\|}{\|V(s_a^{raiz})\|} & \text{se } hits(s_a^{raiz}) < n \\ \frac{hits(s_a)}{hits(s_a^{raiz})} & \text{se } hits(s_a^{raiz}) \geq n \end{cases} \tag{4.11}$$

onde n é o número mínimo de notificações numa amostra significativa. Convém notar que a função $P()$ depende das subscrições submetidas localmente e das notificações recebidas, pelo que poderá ser diferente entre servidores de acesso. Por conseguinte, $P()$ comporta-se como uma função de probabilidade local a cada servidor de acesso. O peso de cálculo deste modelo não é significativo, pois o processo de inserção de uma nova subscrição no grafo, permite determinar tanto s_a^- como s_a^+, pelo que o único custo reside na computação do estimador de $hits(s_a)$. O único requisito de qualidade para $P()$ reside na amostra de emparelhamentos recolhidos ao longo do tempo. Se for necessário, pode-se guardar persistentemente o estado dos contadores do grafo de forma a manter a qualidade da estimação de $P()$ entre paragens do servidor de acesso.

[3]Neste caso a raiz não tem subscritores.

4.7 Peso de subscrição

O algoritmo de emparelhamento usa a função de probabilidade $P()$ na determinação das melhores agregações de subscrições em grupos de difusão. Contudo, só por si o valor de qualquer probabilidade não é suficiente quando se pretende escolher entre subscrições de assuntos diferentes para atribuição a um grupo. Por exemplo, para um certo par de subscrições podemos ter $P(s_a) \approx P(s_b)$ e no entanto os assuntos a e b podem apresentar taxas de emissão de notificações bastante diferentes. Consequentemente, é necessário introduzir o conceito de peso de subscrição $W(s_a)$, o qual sendo baseado na probabilidade de notificação $P(s_a)$, é dado pela fórmula

$$W(s_a) = W(a) \cdot P(s_a), \tag{4.12}$$

onde $W(a)$ representa o peso do assunto a expresso em byte/s ou notificação/s[4].

Por comparação com outras soluções de emparelhamento de subscrições em grupos, o uso da função $W(s_a)$ permite determinar agregações mais eficientes. Tomando o exemplo da Tabela 4.1 e considerando a solução na Tabela 4.2(a), temos que o subscritor C recolhe um excesso de notificações no valor de $W(e)$. Se fosse adoptado um algoritmo de emparelhamento que reutilizasse ciclicamente os grupos após o seu esgotamento (como por exemplo em [7]), e considerando apenas sete grupos disponíveis, teríamos a solução (a) da Fig. 4.2. Com o uso de $W(s_a)$, o algoritmo proposto em seguida pretende obter soluções tal como em (b).

Por fim, podemos definir o custo suportado pelo subscritor m membro do grupo ipm_g por,

[4] $W(a)$ poderá ser fixo ou variar.

Grupo	**Subscrição**	**Subscrição**
ipm_1	$_a(ABG)_i(CD)$	$W_i(ABG)W_a(CD)$
ipm_2	$_b(A)_t(E)$	$W_t(A)W_b(E)$
ipm_3	$_c(BG)_u(F)$	$W_u(BG)W_c(F)$
ipm_4	$_d(A)_v(BG)$	$W_v(A)W_d(BG)$
ipm_5	$_e(D)_x(F)$	$W_x(D)W_e(F)$
ipm_6	$_g(C)_z(E)$	$W_z(C)W_g(E)$
ipm_7	$_h(C)$	

(a)

Grupo	**Subscrição**	**Subscrição**
ipm_1	$_a(ABG))$	
ipm_2	$_{b,d}(A)$	
ipm_3	$_{c,v}(BG)$	
ipm_4	$_{g,h}(C)$	
ipm_5	$_e(D)_i(CD)$	$W_e(C)$
ipm_6	$_{u,x}(F)$	
ipm_7	$_{t,z}(E)$	

(b)

Figura 4.2: Custos incorridos com reutilização simples de endereços (a), e com o algoritmo proposto (b).
Nesta figura, $_i(CD)$ representa a subscrição i efectuada por C e D, e $W_e(C)$ representa o custo $W(e)$ incorrido por C pelo facto de não ter subscrevido e, e e estar agregado a um grupo de difusão do qual C é membro.

$$WG_g(m) = \sum_{a \in A_g} W(\bigcup_g s_{a_k} \setminus s_{a,m}) \tag{4.13}$$

onde A_g é o conjunto de todos os assuntos para os quais existem subscrições em ipm_g, e $\bigcup_g s_{a_k}$ representa a reunião de todas as subscrições parcelares s_{a_k} de a em ipm_g[5].

4.8 Procura genética

Em §4.2 vimos que o algoritmo é matematicamente representado pela relação R em $S \times G$, onde S representa o conjunto das expressões de subscrição e G a lista de grupos

[5] Relembrar em §4.3 a decomposição interna de uma subscrição em várias subscrições disjuntas.

disponíveis. No processo de atribuição de subscrições a endereços de difusão, a relação R procura pela melhor solução de emparelhamento no espaço $S \times G$. Dependendo do sistema em exploração, este espaço de procura pode tomar uma dimensão muito grande, pelo que um algoritmo de pesquisa exaustiva será pouco eficiente ou mesmo inaplicável. Por conseguinte, é necessário recorrer a um algoritmo capaz de encontrar uma solução de emparelhamento, que embora possa não ser a melhor, é suficientemente boa atendendo ao esforço de procura da melhor solução num espaço de grande dimensão. O algoritmo proposto pertence à classe dos algoritmos genéticos [33, 34], e recorre a técnicas que permitem a redução da complexidade do processo de procura. Estas técnicas são semelhantes aos mecanismos biológicos de evolução das espécies, pelos quais partindo de uma população inicial as sucessivas gerações de indivíduos reflectem o bom desempenho dos melhores espécimes das gerações precedentes considerando as opções tomadas pela natureza na selecção, cruzamento e mutação dos indivíduos. Estabelecendo uma analogia com o processo computacional de procura realizado por R, podemos considerar cada solução de emparelhamento como um indivíduo cujo desempenho é avaliado por uma função[6], baseada no custo da recepção de notificações indesejadas.

Tal como listado na Fig. 4.3, o algoritmo genético parte de um conjunto inicial de soluções de emparelhamento (l1), para as quais calcula o respectivo valor de desempenho (l2). Seguidamente, é iniciado um ciclo durante o qual, por aplicação dos operadores genéticos de selecção, cruzamento e mutação, são derivadas novas hipóteses de emparelhamento, sendo sucessivamente escolhidas as soluções de melhor desempenho. No fim, é escolhida a solução que, em função do conjunto de subscrições locais e dos grupos disponíveis, mais reduz a possibilidade de entrega indesejada de notificações. Por sua vez, o processo de procura deverá terminar em tempo útil de execução para não

[6]Designada em inglês por *fitness function*.

l1: [**Início**]
Geração das n soluções (hipóteses de emparelhamento) que irão constituir a população inicial.

l2: [**Desempenho**]
Cálculo do desempenho de cada solução.

l3: [**Aceitação**]
Colocação na população.

l4: [**Teste**]
Termina e retorna a melhor solução se a condição de paragem do algoritmo for satisfeita.

l5: [**Nova população**]
Criação da nova geração de soluções pela repetição dos seguintes passos até nova população estar completa:

l5.1: [**Selecção**]
Selecção de duas soluções da população de acordo com o seu desempenho (quanto melhor for este, maior a probabilidade de ser escolhida).

l5.2: [**Cruzamento**]
Cruzar as duas soluções seleccionadas consoante a probabilidade de cruzamento, criando novas soluções candidatas. Se não ocorrer cruzamento, as novas soluções são uma cópia das primeiras.

l5.3: [**Mutação**]
Introduzir uma mutação nas novas soluções de acordo com a probabilidade de mutação.

l6: Volta a l2.

Figura 4.3: Algoritmo genético.

comprometer a capacidade de escala e o correcto funcionamento do sistema.

Considerando a natureza do problema e a necessidade de encontrar uma boa solução de emparelhamento, existem duas formas principais de agregar subscrições num mesmo grupo de difusão:

- pela semelhança entre os respectivos conjuntos de subscritores;
- pela semelhança entre volumes subscrevidos, identificada pela relações de cobertura $\subset_s^s$ entre as subscrições dum grafo. Sendo $s_a \not\subset_s^s s_b$ sempre válida, esta forma de agregação só é aplicável entre subscrições dum mesmo assunto.

Cabe ao algoritmo, através da aplicação dos operadores genéticos e conduzido pela função de desempenho, encontrar uma solução de emparelhamento baseada numa combinação destas duas formas de agregação. De um modo geral, quanto maior for a semelhança entre as subscrições efectuadas, tanto no que se refere à proximidade dos volumes de subscrição como à semelhança dos conjuntos de subscritores, menos grupos serão necessários. No limite, se todos subscreverem os mesmos volumes, basta só um grupo para distribuição de todas as notificações.

4.8.1 Codificação

Num algoritmo genético cada indivíduo da população deve conter informação relativa à solução que representa dentro do espaço de procura (o seu material genético). Considerando o problema do emparelhamento de subscrições em grupos de difusão, cada indivíduo deve listar para cada grupo as subscrições associadas, tal como mostra a figura seguinte.

solução K								
ipm1	*s1*				*s5*		*s7*	
ipm2			*s3*			*s6*		*s8*
...	...							
ipmN		*s2*		*s4*				

Figura 4.4: Codificação de uma solução de emparelhamento.

Nesta figura, cada grupo é representado por uma linha onde se listam ordenadamente as subscrições que lhe foram atribuídas. Os espaços em branco dizem respeito a subscrições associadas a outros grupos. Como veremos adiante, a operação de cruzamento é facilitada pela simplicidade desta codificação e pela ordenação das subscrições dentro de cada grupo.

4.8.2 População inicial

A escolha de boas soluções de emparelhamento na constituição da população inicial pode impulsionar o processo de procura e contribuir para o encurtamento do tempo de execução do algoritmo. A ideia resume-se em começar por soluções de desempenho próximo do desempenho das melhores soluções do espaço de procura. Neste sentido, a população inicial pode ser gerada tendo em conta as duas formas de agregação de subscrições atrás.

Uma hipótese imediata consiste em agregar num mesmo grupo as subscrições dum assunto, num grupo seguinte as subscrições de outro assunto, e assim por diante. Esta solução tende a tirar partido da intersecção entre volumes subscrevidos dum mesmo assunto. Outra possibilidade, com eventuais aplicações em sistemas cooperativos, jogos distribuídos, e de um modo geral, em sistemas de difusão de informação com cariz geográfico, consiste em agrupar num mesmo grupo as subscrições de clientes ”suficientemente perto”[7] entre si. Por fim, uma outra hipótese consiste em atribuir todas as subscrições a um único grupo, formado uma solução de difusão total (*flooding*).

4.8.3 Cálculo do desempenho

A qualidade duma solução de emparelhamento é medida pela seguinte função de custo,

$$C(emp) = \sum_{g=1}^{G} C_g(emp) \tag{4.14}$$

onde $C_g(emp)$ é dado por

$$C_g(emp) = \sum_{m_{g,k} \in M_g} WG_g(m_{g,k}). \tag{4.15}$$

[7] A percepção/quantificação de ”suficientemente perto” dependerá de cada aplicação.

Nesta fórmula, G é o número de grupos utilizados no emparelhamento emp, e M_g é o conjunto dos subscritores $m_{g,k}$ associados ao grupo ipm_g. Deste modo, quanto menor for o custo duma solução de emparelhamento, melhor o seu desempenho, e maior a hipótese de ser escolhida como solução final.

4.8.4 Selecção

A selecção consiste na escolha de um par de soluções da população corrente, com vista ao seu cruzamento, tendo em consideração o seu melhor desempenho relativamente às restantes soluções. Contudo, a escolha estrita de indivíduos com bom desempenho pode conduzir a uma especialização bastante rápida da população de soluções nas iterações iniciais do algoritmo, e consequentemente à estagnação deste num mínimo local da função de custo, $C(emp)$. Como tal, o mecanismo de selecção deverá também dar oportunidade a soluções de desempenho inferior, pelo menos nas fases iniciais de execução. Pretende-se assim aumentar a diversidade das gerações iniciais, o que se traduz num maior alcance dentro do espaço de soluções possíveis.

A este respeito, o algoritmo realizado adopta inicialmente um mecanismo de selecção por torneio (do inglês, *tournament selection*), e numa fase mais adiantada da pesquisa um mecanismo de roleta (do inglês, *roulette wheel selection*), promovendo assim a especialização das soluções nessa altura.

4.8.5 Cruzamento

A operação de cruzamento envolve a troca de material genético entre pares de indivíduos, na esperança dos indivíduos resultantes reunirem as melhores características dos seus progenitores. Relativamente ao problema em causa, o cruzamento de um par de soluções é concretizado pela troca de sequências de subscrições entre grupos de cada solução. A determinação destas sequências de subscrições é facilitada pela codificação proposta. Antes do cruzamento ocorrer é escolhido um par de subscrições como pontos de cruzamento[8]. Uma vez que as subscrições são ordenadas por identificador dentro de cada grupo, os pontos de cruzamento definem os limites das sequências de subscrições a trocar.

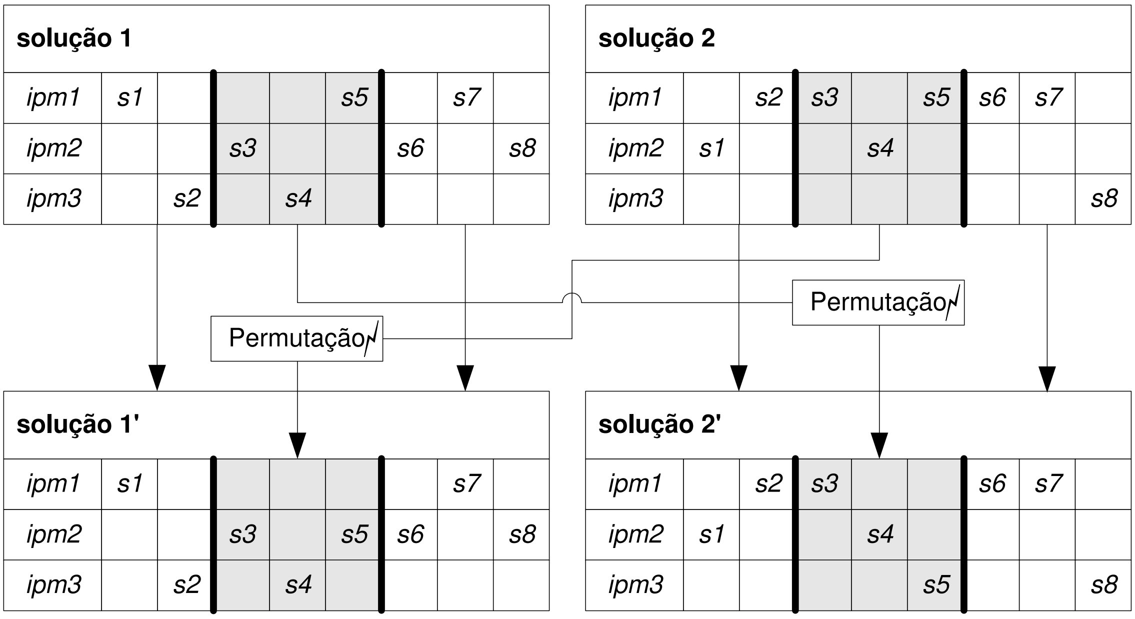


Figura 4.5: Cruzamento entre as soluções 1 e 2 resulta nas soluções 1′ e 2′. Os traços a negro representam os pontos de cruzamento $s3$ e $s5$.

Assim, tal como mostra a figura, cada nova solução deriva de duas soluções ascendentes da seguinte forma: à esquerda e à direita pelas subscrições à esquerda e à direita, respectivamente do primeiro e segundo pontos de cruzamento, numa das soluções; e ao centro, por uma permutação aleatória das sequências de subscrições dentro dos dois pontos de cruzamento, na outra solução. Note-se que a partir de um par de soluções iniciais resulta sempre um novo par de soluções candidatas.

Convém referir que a codificação definida apresenta uma propriedade bastante im-

[8]O par é formado na realidade pelos identificadores das subscrições escolhidas.

portante em resultado da ordenação imposta às subscrições em cada grupo: para qualquer par de soluções válidas o cruzamento destas resulta num par de soluções igualmente válidas, nunca ocorrendo perda ou repetição de subscrições por cruzamento. Por fim, apesar do exemplo anterior demonstrar o uso de dois pontos de cruzamento, a codificação suporta facilmente outras formas de cruzamento, como sejam as de múltiplos pontos de cruzamento.

4.8.6 Mutação

A operação de mutação traduz-se pela alteração do material genético dum indivíduo, tendo como objectivo retirá-lo de uma posição de óptimo local para a qual tenha escorregado o seu desempenho. Pretende-se assim potenciar saltos dos indivíduos da geração actual para posições mais favoráveis dentro do espaço de procura.

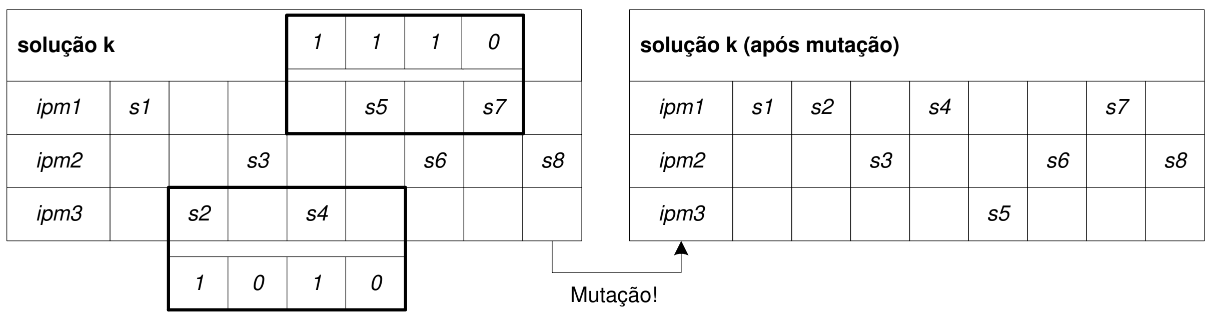


Figura 4.6: Mutação de uma solução.

Observando a figura, a mutação é realizada pela troca de subscrições entre um par de grupos da solução. Em cada grupo seleccionado, as subscrições a trocar são escolhidas de acordo com uma máscara binária aleatória.

Eventualmente, através da mutação as soluções podem ser melhoradas pela libertação de grupos pouco ocupados. Tal como atrás, a aplicação desta operação a uma solução válida resulta noutra solução válida.

4.8.7 Aceitação

A aceitação assenta na inserção das novas soluções na próxima população, de acordo com os desempenhos obtidos. Contudo, neste processo podem ser perdidas soluções com bom desempenho pertencentes à população corrente. Por conseguinte, as melhores soluções desta são também copiadas para a nova população. Este processo de retenção das melhores soluções da população corrente é denominado de elitismo (do inglês *elitism*).

4.8.8 Teste de fim do algoritmo

O algoritmo de procura deverá terminar após um número de iterações predefinido ou quando for alcançado um melhoramento significativo em relação ao emparelhamento actual. Por fim, a solução retornada é a de melhor desempenho. Se os melhores desempenhos forem iguais é retornada a solução que utilizar menos grupos.[9]

4.8.9 Iniciação da procura

O algoritmo de emparelhamento é executado quando é recebido um pedido de subscrição ou de cancelamento, ou quando o número de grupos de difusão disponíveis aumenta ou diminui. Contudo, nem sempre que é executado o algoritmo é realizada uma procura genética, sob pena de se reduzir bastante a capacidade de escala. As subscrições consideradas para entrega de notificações por difusão são aquelas em que o número de subscritores excede um certo limite, abaixo do qual as notificações são entregues ponto-a-ponto. Se este limite for ultrapassado por um novo pedido de subscrição, o algoritmo determina o melhor grupo a devolver aos respectivos subscritores, de acordo com o de-

[9] Antes de ser usada uma nova configuração de emparelhamento é necessário comunicá-la aos subscritores. Uma hipótese poderá passar pelo envio de uma mensagem de mudança de configuração, para um ou mais grupos em uso pelos subscritores interessados.

sempenho resultante da atribuição de um grupo livre ou já ocupado. Se porventura de um pedido de subscrição resultar um aumento de volume de subscrição, quem subscreve pode verificar uma diminuição de notificações entregues indevidamente, enquanto que os restantes subscritores associados ao mesmo grupo podem notar um aumento. Tratando-se de uma diminuição do volume de subscrição pode acontecer o efeito contrário. Se por configuração do sistema diminuir o número de grupos disponíveis, as subscrições associadas a esses grupos têm de ser redistribuídas pelos restantes. Em qualquer destes casos o desempenho do emparelhamento obtido é calculado, e se for um certo valor abaixo da média dos últimos[10], então é iniciado o processo genético de procura por um melhor emparelhamento. Por outro lado se aumentar o número de grupos livres, quer por libertação de grupos ocupados quer por disponibilização de novos grupos, uma nova procura poderá ser feita para diminuir o custo nos grupos ocupados. Note-se que a alteração de disponibilidade de grupos de difusão por configuração não é uma operação que deva ocorrer com frequência. Para finalizar esta secção, o desempenho do emparelhamento utilizado deve ser calculado periodicamente se o peso de subscrição $W(a)$ de cada assunto variar dinamicamente. Com base na actualização do valor de desempenho, deve ser tomada a decisão de se procurar ou não por um novo emparelhamento.

4.9 Optimizações

Nesta secção é listado um conjunto de melhoramentos ao algoritmo apresentado. Tratando-se de optimizações a sua concretização é apenas aconselhada. Estas melhorias focam vários aspectos do algoritmo tais como, a sua capacidade de escala, o consumo de memória primária, e o tempo de processamento.

[10]Consideram-se os últimos n emparelhamentos, uma vez que correspondem a estados do sistema mais próximos do actual.

4.9.1 Segmentação do grafo de coberturas

Para cada assunto as subscrições recebidas podem produzir um grafo mal equilibrado se não for possível estabelecer entre elas suficientes relações de cobertura. Numa situação dessas, a introdução estratégica de *nós de segmentação* pode restabelecer o equilíbrio do grafo, e tornar menos pesada a determinação do conjunto de subscritores a entregar uma notificação. Estes nós de segmentação de grafo não são agrupados em grupos de difusão mas podem ajudar no processo de criação da população inicial de soluções na procura genética. Note-se, que a própria raiz de cada grafo é por vezes um nó de segmentação, pois pode muito bem suceder que seja apenas uma cobertura das subscrições realmente submetidas nos nós imediatamente abaixo.

4.9.2 Descarregamento de sub-grafos

Da mesma forma como o grafo dum assunto pouco frequente pode estar guardado em disco, um sub-grafo abaixo de um nó (de segmentação ou outro) pouco visitado (poucos *hits*) também poderá ser guardado em disco e ser só chamado em caso de necessidade. Entretanto, se for recebido um pedido de subscrição coberto pelo nó raiz dum sub-grafo em disco, esse pedido de subscrição pode ser internamente diferido para quando surgir uma notificação que emparelhe com esse nó. Nessa altura este é lido do disco e os pedidos de subscrição pendentes são executados. A escolha dos sub-grafos para armazenamento em disco pode ser feita explicitamente, ou implicitamente pelo mecanismo de memória virtual do sistema operativo. Em qualquer destes casos, é necessário marcar o nó raiz dos sub-grafos em questão.

4.9.3 União de nós de subscrição

Eventualmente a quantidade de nós num grafo é tão grande que atrasa o encaminhamento de notificações bem como o emparelhamento de subscrições em grupos. Com uma grande probabilidade muitos desses nós correspondem a volumes de subscrição muito próximos entre si. Nestes casos, a substituição de vários nós por um nó cuja expressão de subscrição seja a união das expressões de subscrição dos primeiros (em inglês, *subscription merging*), torna mais rápido o encaminhamento das notificações recebidas, reduz o número de subscrições a considerar na procura de emparelhamentos, e alivia a estrutura de informação do grafo. De um modo geral, a união de nós contribui para o aumento da capacidade de escala do sistema. A união pode ser realizada entre nós que partilhem um mesmo nó de cobertura e inclusivamente com este, desde que a proporção de volume comum seja superior a uma certa percentagem. Este processo é recursivo pois vários nós de união podem ser fundidos noutros nós. Contudo, a diferença de volumes entre qualquer par de nós reunidos não pode ultrapassar o valor estabelecido[11]. A criação de uniões entre nós pode acontecer sempre que é processado um novo pedido de subscrição ou então periodicamente por consulta do grafo. No entanto, só deve ser iniciada após um número de notificações suficiente, pois para além da proximidade entre os seus volumes, dois nós só devem ser reunidos se a diferença entre os seus contadores (*hits*) também for pequena. Caso contrário não só se pode perder informação estatística relevante, como a entrega indevida de notificações pode aumentar para os subscritores do nó com menos *hits*. Por fim, convém salientar o facto desta optimização só poder ser realizada nos servidores de acesso e nunca nas máquinas onde residem os processos subscritores, uma vez que estes não podem receber notificações indesejadas.

[11]Para garantir esta proximidade é necessário e suficiente guardar sempre o menor e o maior dos volumes reunidos.

4.9.4 Comutação entre modos de entrega

Para qualquer subscrição, o número de subscritores varia ao longo do tempo de acordo com factores não controláveis, externos ao sistema de publicação. Este comportamento pode causar sucessivas mudanças na forma de entrega das notificações, entre os modos ponto-a-ponto e difusão, causando instabilidade no sistema. Por conseguinte, para além do limite superior de subscritores para entrega de notificações ponto-a-ponto, é introduzido um limite inferior para continuação da entrega por difusão, abaixo do qual as notificações voltam a ser distribuídas ponto-a-ponto. Desta forma diminui-se o impacto dalguma instabilidade exterior.

4.9.5 Filtragem de subscrições de peso reduzido

Algumas subscrições podem ter um número de subscritores suficiente para que sejam distribuídas por difusão, mas o respectivo peso de subscrição é bastante pequeno. Nestes casos, a carga de notificações que estes subscritores poderiam receber por necessidade de adesão a um grupo, é eliminada, se as poucas notificações forem entregues ponto-a-ponto. Consequentemente, estas subscrições não são consideradas para atribuição a grupos, pelo que se reduz o número de subscrições na procura de agrupamentos.[12]

4.10 Sumário

Neste capítulo apresentámos um novo algoritmo aplicável a sistemas de publicação e subscrição de informação utilizando *IP Multicast*. Este algoritmo realiza o emparelhamento de subscrições de conteúdo em grupos. Introduzimos também os conceitos de

[12] Eventualmente todas as subscrições nestas condições podem ser agrupadas num grupo especial, em vez de serem distribuídas ponto-a-ponto.

volume de subscrição, probabilidade de notificação, e de peso de subscrição, fundamentais ao algoritmo proposto. Este último está na base da estimativa da quantidade de notificações entregues indevidamente a um subscritor durante a procura de arranjos de subscrições em grupos de difusão. A motivação para o desenvolvimento deste algoritmo deriva das vantagens da utilização do *IP Multicast* na realização de sistemas de publicação na *Internet*, em particular, a possibilidade de envio de um único pacote para um grupo de subscritores. Nesse sentido, o algoritmo descrito propõe uma solução para ultrapassar os desafios que a concretização em *IP Multicast* impõe. Também vimos que qualquer solução de agrupamento de subscrições pode ser avaliada pelo número de grupos usados, pelo custo das notificações não esperadas, e pelo peso de execução. Na solução proposta é utilizado um algoritmo genético para pesquisa de um agrupamento tendo em conta estes critérios. Um algoritmo genético impõe-se pela grandeza do espaço de soluções possíveis. Por fim, apresentámos um conjunto de optimizações à solução base.

Capítulo 5

Simulação do algoritmo de procura genética

Este capítulo apresenta uma avaliação do algoritmo de emparelhamento através de simulação, dado que este é um componente chave da arquitectura, e uma das principais contribuições desta dissertação. O desenvolvimento de um protótipo da arquitectura na sua totalidade está fora do âmbito deste trabalho, sendo deixado para trabalho futuro.

5.1 Ferramenta utilizada

Toda a simulação foi desenvolvida recorrendo à ferramenta MATLAB 6.0. A escolha desta ferramenta deveu-se ao facto de disponibilizar uma linguagem interpretada, permitindo rapidamente concretizar, testar e afinar o algoritmo de emparelhamento proposto. Simultaneamente, o MATLAB 6.0 oferece um vasto conjunto de primitivas matemáticas, das quais se destacam as de manipulação de vectores e matrizes, o que reduz o tempo de escrita do algoritmo; bem como facilidades de produção de gráficos a partir dos resultados obtidos na simulação.

Posteriormente à codificação do algoritmo, foi necessário reescrever algumas das suas funções em linguagem C, em virtude da lentidão quando interpretadas em MATLAB 6.0. Nessa altura, a ferramenta mostrou ser bastante flexível, suportando a integração de módulos escritos em C através da utilização da *MEX API* (*MEX Application Programming Interface*).

De um modo geral, o MATLAB 6.0 revelou-se como uma excelente ferramenta de prototipagem e avaliação de algoritmos genéticos.

5.2 Representação do espaço de subscrição

Na simulação desenvolvida, o espaço de informação dum assunto é representado por uma sequência de volumes atómicos (disjuntos, indivisíveis, e de diferentes dimensões), onde cada um representa uma parcela do volume total do assunto.

Cada subscrição é então formada por um subconjunto de volumes atómicos consecutivos. Este subconjunto é determinado, começando por escolher aleatoriamente um dos volumes do assunto, e depois os volumes $-r$ à esquerda e $+r$ à direita do volume inicialmente escolhido, onde r é o *raio de subscrição*. No total, cada subscritor pode fazer até k subscrições, sendo no final considerada a união destas, ou seja, a união dos volumes atómicos em cada assunto subscrevido.

Esta representação do espaço de informação não limita as conclusões derivadas desta simulação, pois os assuntos podem ser divididos em tantos volumes quantos se queiram, suportando desse modo, cenários com subscrições mais dispares entre si. Contudo, tem a vantagem de simplificar bastante o cálculo da Eq. 4.13, essencial no cálculo da função de custo (Eq. 4.14).

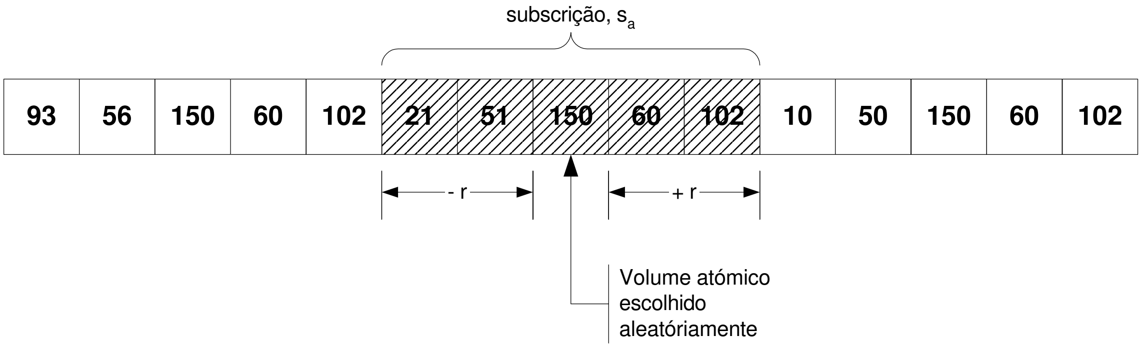


Figura 5.1: Volumes atómicos dum assunto e volumes duma subscrição. Cada número indica a dimensão relativa do respectivo volume dentro do assunto. As dimensões são atribuídas aleatoriamente, variando entre a unidade e um limite máximo.

5.3 Processo de simulação

O processo de simulação tem por base a utilização de ficheiros de configuração de testes. Cada ficheiro de configuração lista um conjunto de testes a serem executados sequencialmente.

```
disp('Teste: número de categorias de assuntos ...');
runtest(0, 100, 1, 0, 64, 255, [20:20:160], 0.1, 10, 20, 4, 8, 500, 40, 16, 5, 0.9, 0.2, 0, 0.4, 0.7); beep;
runtest(0, 100, 5, 0, 64, 255, [20:20:160], 0.1, 10, 20, 4, 8, 500, 40, 16, 5, 0.9, 0.2, 0, 0.4, 0.7); beep;

disp('Teste: percentagem de assuntos com interesse ...');
runtest(0, 200, 5, 4, 32, 1, [20:20:160], 0.1, 32, 32, 1, 1, 500, 40, 16, 10, 0.9, 0.2, 0.1, 0.4, 0.7); beep;
runtest(0, 200, 5, 4, 32, 1, [20:20:160], 0.05, 32, 32, 1, 1, 500, 40, 16, 10, 0.9, 0.2, 0.1, 0.4, 0.7); beep;
```

Figura 5.2: Exemplo de ficheiro de teste.
As funções *disp()* e *beep* não têm qualquer importância na realização dos testes, servindo apenas para indicar progresso na execução destes.

Estes ficheiros são executados através do guião ***runscript(<fich-teste>)***, o qual procura no directório 'tests\' pelo ficheiro de configuração chamado <*fich-teste*>. No ficheiro de configuração, cada teste é realizado pela invocação do guião ***runtest()***, o qual toma os parâmetros listados na Tab. 5.1.

(a)	
bIsSpatial	0 (reservado para uso futuro)
nsubjects	número de assuntos
nsbjcategories	número de categorias de assuntos
maxsbj_power	máximo da taxa de tráfego relativa dum assunto (potência de 10)
natoms	número de volumes atómicos
maxatom_weigth	máximo da dimensão dum volume atómico
atest	lista de configurações de subscritores (ex: 10, 20, 50 subscritores)
psubjects	percentagem de assuntos a subscrever por cada subscritor
nminradius	mínimo do raio de subscrição
nmaxradius	máximo do raio de subscrição
nminsubspercli	número mínimo de subscrições por subscritor
nmaxsubspercli	número máximo de subscrições por subscritor
(b)	
nMaxIteration	número de iterações do algoritmo genético
npopsize	tamanho da população
ngroups	número de grupos disponíveis
useElitism	número de soluções elitistas
pCrossover	probabilidade de cruzamento
pMutation	probabilidade de mutação
pSwitchCrossover	percentagem de iterações iniciais usando cruzamento aleatório
pSwitchSelection	percentagem de iterações iniciais usando selecção por torneio (*tournament selection*)
pTournament	probabilidade da melhor solução do torneio ser seleccionada

Tabela 5.1: Parâmetros do guião *runtest()*.
Os grupos (a) e (b) de parâmetros dizem respeito ao cenário de subscrição e ao algoritmo genético, respectivamente.

O processo de simulação (Fig. 5.3) adopta uma filosofia de reaproveitamento de cálculos intermédios anteriormente realizados (*cache*), de modo a reduzir o tempo de execução em testes futuros. Estes cálculos são guardados numa estrutura de ficheiros, debaixo da pasta 'runs\'. Consoante o tipo de informação a que respeitam, encontram-se organizados tal como mostra a Tab. 5.2. Note-se nesta tabela, que ficheiros dum mesmo tipo distinguem-se através da concatenação da cadeia de parâmetros de entrada para a função a que respeitam.

$runs\backslash\!<\!c_k\!>\!\backslash cdata.mat$	cenário de subscrição c_k (***runcenary()***)
$runs\backslash\!<\!c_k\!>\!\backslash genesis\_<\!p_x\!>.mat$	população inicial p_x em c_k (***rungenetic()***)
$runs\backslash\!<\!c_k\!>\!\backslash genetic\_<\!g_y(p_x)\!>.mat$	resultado da pesquisa genética g_y em c_k, tomando p_x como população inicial (***rungenetic()***)
$runs\backslash test\_<\ldots>.mat$	resultado de uma amostra de notificações produzida para um conjunto de cenários c_k considerados para teste (***runtest()***)
$runs\backslash history\backslash teste\_i.mat$	histórico dos resultados dos testes do ficheiro 'teste_i.m' (***runscript()***)

Tabela 5.2: Estrutura de ficheiros derivados do processo de simulação.

5.3.1 Mecanismos de extracção de resultados

De forma a facilitar a visualização dos resultados do processo de simulação, foram escritos alguns guiões que permitem listar dados e traçar gráficos dos mesmos (Tab. 5.3). Estes guiões, aliados às funcionalidades gráficas da ferramenta utilizada, permitiram facilmente produzir gráficos dos resultados dos testes efectuados, apresentados na secção seguinte.

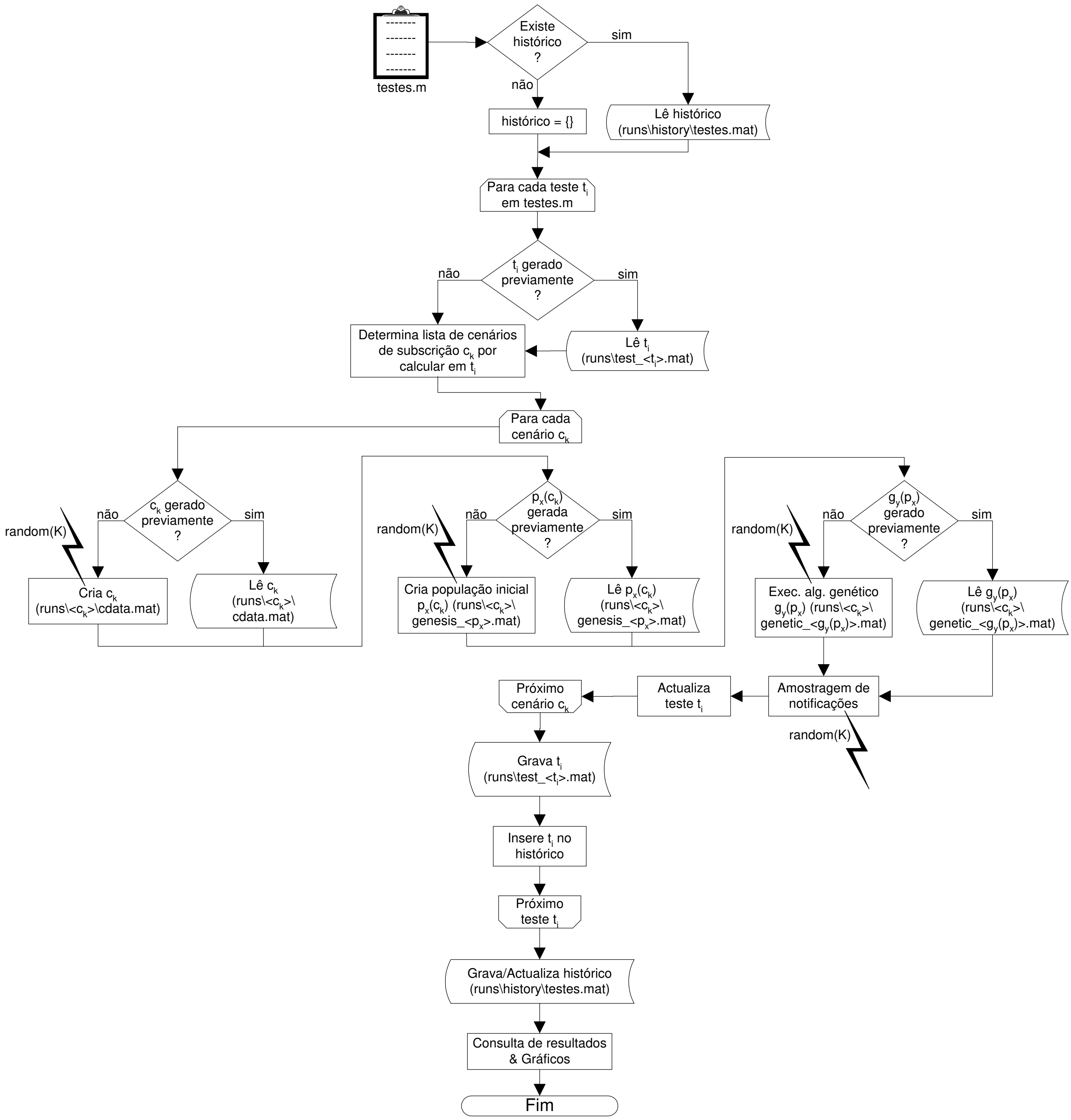


Figura 5.3: Fluxograma do processo de simulação.

Cada ficheiro de teste lista vários testes (chamadas a ***runtest()***), e cada teste define um conjunto de cenários de subscrição c_k (diferindo apenas no número de clientes). Para cada cenário é gerada uma população inicial $p_x(c_k)$, após o que é executado o algoritmo genético, gerando dados $g_y(p_x)$.

findgen()	Permite procurar pelos resultados de uma pesquisa genética em função do desempenho das soluções encontradas. Ex: $findgen('teste\_a','v > 0.5')$, retorna todos os resultados genéticos no histórico 'teste_a.mat' onde a melhoria de desempenho foi superior a 50%.
dispgen()	Lista todos os resultados genéticos num ficheiro de histórico de testes. Ex: $dispgen('teste\_a')$.
plotgen()	Traça o gráfico dos resultados de uma ou mais pesquisas genéticas num mesmo histórico, para comparação. Ex: $plotgen('teste\_a', [1\ 4])$, permite comparar os resultados do 1º e 4º testes de procura genética no histórico 'teste_a.mat'. Este guião aceita como terceiro argumento uma *string* com os nomes dos gráficos a traçar, separados por vírgulas: $'f'$, custo da melhor solução; $'std'$, desvio padrão da população; $'gap'$, soluções novas em cada iteração (*generation gap*). Por defeito o gráfico traçado corresponde a $'f'$.
disptest()	Lista todos os testes presentes num histórico, ou então os detalhes dum teste nesse histórico. Ex: $disptest('teste\_a')$, lista todos os testes guardados no histórico 'teste_a.mat', e $disptest('teste\_a', 3)$, apresenta os resultados da amostragem de notificações do 3º teste em 'teste_a.mat'.
plottest()	Traça o gráfico dos resultados de um ou mais testes num mesmo histórico, para comparação. Ex: $plottest('teste\_a', [2\ 3])$, permite comparar a percentagem de mensagens correctamente entregues em cada cenário, no 2º e 3º testes. Este guião aceita como terceiro argumento uma *string* com os nomes dos gráficos a traçar, separados por vírgulas: $'s'$, número de subscrições distintas; $'su'$, número de subscrições servidas ponto-a-ponto (*unicast*); $'sm'$, número de subscrições servidas por difusão (*multicast*); $'u'$, número de not. enviadas ponto-a-ponto; $'[f\|0\|]m'$, número de not. enviadas por difusão, na solução de difusão total (f, *flooding*), solução inicial (0), ou solução final (vazio), respectivamente; $'[f\|0\|]ok'$, número de not. bem entregues por difusão; $'[0\|]ok2'$, número de 'ok' em duplicado; $'[f\|0\|]bad'$, número de not. mal entregues por difusão; $'[0\|]bad2'$, número de 'bad' em duplicado; $'[f\|0\|]good'$, percentagem de not. bem entregues $(= ok/(bad + ok2 + bad2)$; $'[0\|]mr'$, número de not. repetidas e enviadas por difusão; $'ou'$, número de not. a emitir numa solução exclusivamente por ponto-a-ponto. Por defeito o gráfico traçado corresponde a $'good'$.

Tabela 5.3: Lista de guiões para visualização dos resultados dos testes.

5.4 Resultados

Esta secção mostra os resultados dos testes efectuados durante a simulação. Inicialmente, procurou-se determinar boas combinações de valores para os parâmetros do algoritmo genético. Assim, os gráficos iniciais dizem respeito à influência destes parâmetros na escolha de boas soluções de emparelhamento. Os restantes gráficos mostram a qualidade das soluções de emparelhamento obtidas em cada cenário testado.

Em todos os testes realizados, os assuntos foram agrupados em uma ou cinco categorias, e cada cliente efectuou subscrições exclusivamente de assuntos duma única categoria, escolhida aleatoriamente. Os assuntos da categoria seleccionada foram escolhidos de acordo com uma distribuição uniforme. Em cada teste, todos os clientes subscreveram a mesma quantidade de assuntos.

Todas as subscrições em por-assunto foram consideradas para difusão. Por seu lado, atendendo à maior quantidade de subscrições em por-conteúdo, estas foram ordenadas em função do seu peso de subscrição, e somente as 70% com maior peso foram consideradas pelo algoritmo de emparelhamento, para entrega por difusão. As restantes foram servidas directamente aos respectivos subscritores.

A população inicial de soluções foi constituída por uma solução de difusão total (um único grupo de difusão), por várias soluções de emparelhamento das subscrições de cada assunto em grupos alternados (*round-robin*), e por várias soluções de emparelhamento indiscriminado. Em todos os testes o número de iterações do algoritmo genético foi de 500.

O raio de subscrição variou aleatoriamente entre dois valores definidos. Em por-assunto estes dois valores igualaram o número de volumes atómicos[1]. Em por-conteúdo o número de volumes atómicos de cada assunto foi de 64 (com tamanhos relativos entre

[1]Neste caso 32 volumes atómicos de tamanho 1.

1 e 255), e o raio de subscrição variou entre 10 e 20 volumes, tendo cada cliente efectuado 4 a 8 subscrições.

Em geral, os restantes valores dos parâmetros de teste, foram: 100 ou 200 assuntos; 1 ou 5 categorias; taxas de tráfego relativas até 10^4; 20 a 160 subscritores; subscrição de 5% ou 10% do total de assuntos; 10, 20, 40 ou 60 soluções iniciais; 16, 32 ou 64 grupos; 0, 1, 5 ou 10 soluções elitistas; probabilidade de cruzamento de 0.4, 0.6, 0.8 ou 0.9; probabilidade de mutação de 0.05, 0.1, 0.2 ou 0.3; cruzamento com permutação aleatória até 10% das iterações iniciais; selecção por torneio até 40% ou 60% das iterações iniciais; e, probabilidade de selecção da melhor solução no torneio de 0.3, 0.7 ou 0.9.

No final, a qualidade duma solução de emparelhamento foi medida pela proporção de notificações com interesse relativamente ao total de notificações recebidas, por todos os clientes. Em todos os testes foi simulado o envio de 1000 notificações, tendo sempre em consideração a taxa de emissão de notificações de cada assunto.

5.4.1 Testes dos parâmetros genéticos

Seguidamente são apresentados gráficos relativos à execução do algoritmo genético. Nestes, o eixo xx representa o número de iterações e o eixo yy o custo da melhor solução no momento. No título de cada gráfico, cC identifica cenários de teste com C categorias de assuntos, pP taxas de tráfego relativas até 10^P, e gG o uso de G grupos de difusão.

Efeito da alteração dos parâmetros genéticos, em por-assunto

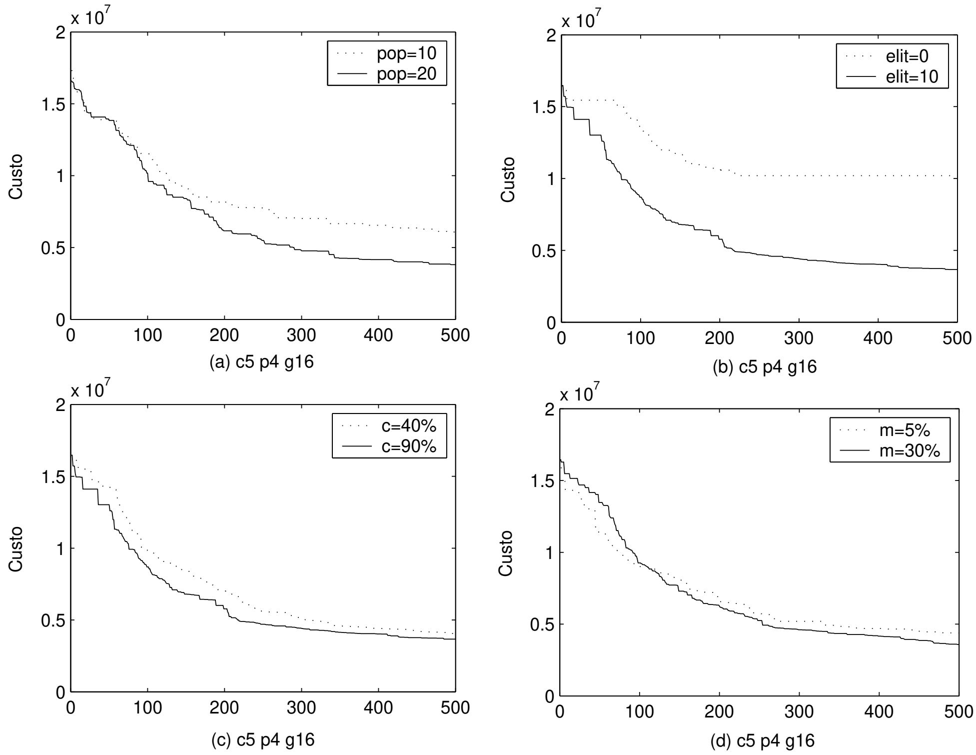


Figura 5.4: Efeito da alteração dos parâmetros genéticos, em por-assunto.

Nestes testes foram considerados 160 subscritores. Foram divididos 100 assuntos em cinco categorias, e cada cliente subscreveu 10 assuntos entre os 20 da categoria escolhida. Foi considerada uma taxa de tráfego relativa até 10^4, e nos emparelhamentos foram utilizados 16 grupos. Em todos os testes, foi feita a variação de um único parâmetro, tendo os restantes sido fixados nos melhores valores determinados.

Verificou-se que o aumento da população de 10 para 20 elementos, resultou num aumento da percentagem de notificações correctamente entregues de 44% para 56%. Acima de 20 elementos, as melhorias não foram significativas. Verificou-se também que o uso de elitismo é bastante importante, obtendo-se melhorias significativas na percentagem de notificações correctas de 32% para 53%. Nos testes efectuados, para valores de elitismo superiores a 1, as melhorias não foram além de 3%. A variação da probabilidade de cruzamento (entre 40% e 90%) resultou apenas no melhoramento de 1% na qualidade das soluções obtidas. Por fim, a variação da mutação (entre 5% e 30%) resultou numa variação de 6% na qualidade final.

Efeito da alteração dos parâmetros genéticos, em por-conteúdo

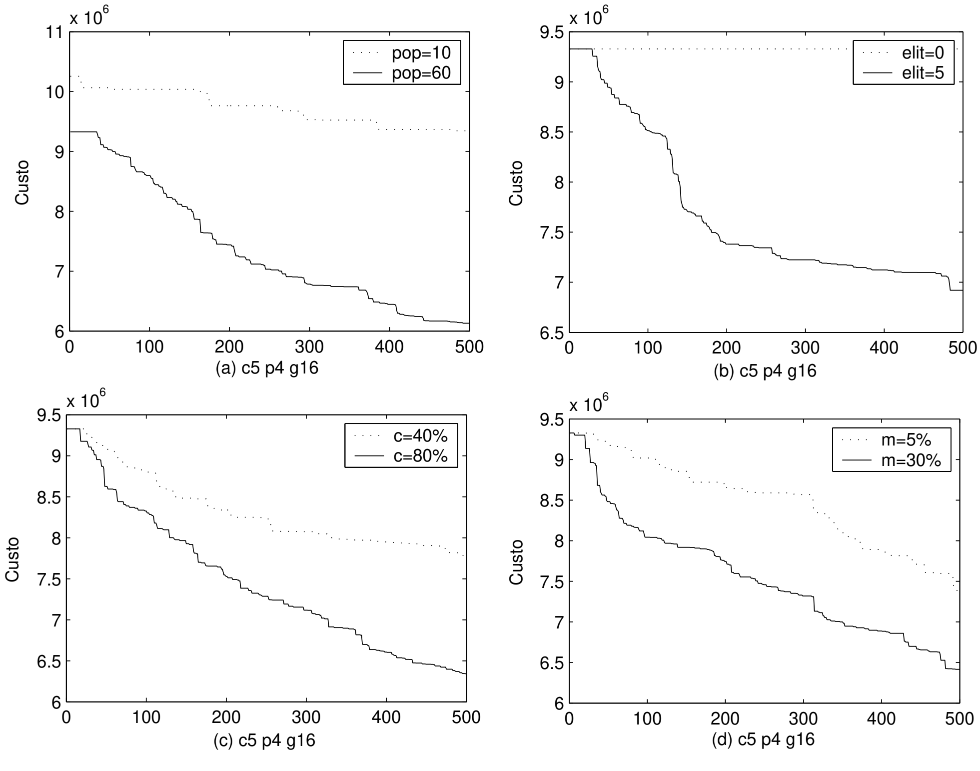


Figura 5.5: Efeito da alteração dos parâmetros genéticos, em por-conteúdo.

Em cenários idênticos, mas com subscrições em por-conteúdo, o aumento do tamanho da população resultou em melhorias na percentagem de notificações correctamente entregues, de 25% com uma população de 10 elementos, para 35% com uma população de 60 elementos. O uso de elitismo também neste caso resultou em melhorias consideráveis, tendo-se passado de 25% sem elitismo, para 35% com 5 soluções elitistas. A alteração da probabilidade de cruzamento (entre 40% e 80%) conseguiu melhorar o emparelhamento em 5%. A variação da probabilidade de mutação (entre 5% e 30%) influenciou positivamente em 2%.

5.4.2 Testes de qualidade das soluções

Seguidamente são apresentados gráficos relativos à qualidade das soluções de emparelhamento. Nestes gráficos, o eixo xx representa o número de subscritores testados, e o eixo yy a proporção de notificações correctamente entregues. No título de qualquer gráfico, cC identifica cenários de teste com C categorias de assuntos, pP taxas de tráfego relativas até 10^P, e gG o uso de G grupos de difusão.

Comparação das soluções de difusão total, emparelhamento alternado de assuntos, e genética, em por-assunto

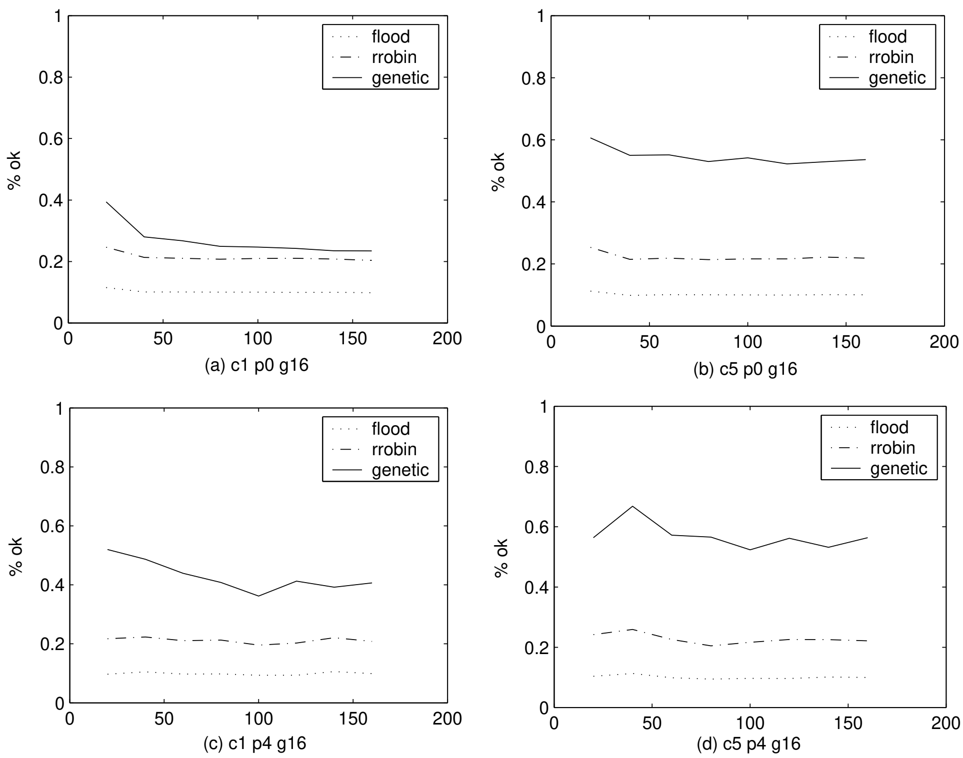


Figura 5.6: Qualidade de diferentes formas de emparelhamento, em por-assunto.

Nestes testes foram considerados entre 20 e 160 subscritores. Em (a) e (c) os 100 assuntos pertenceram a uma única categoria, e em (b) e (d) foram divididos em cinco categorias. Em qualquer dos casos, cada cliente subscreveu 10 assuntos entre os da categoria escolhida. Em (a) e (b) não foram consideradas taxas de tráfego diferentes, e em (c) e (d) a taxa de tráfego relativa variou entre 1 e 10^4. Continuaram a ser utilizados 16 grupos.

Como seria de esperar, verifica-se que a solução de difusão total apresenta sempre os piores resultados, seguida da solução de emparelhamento alternado, e da genética. Tomando estas duas últimas soluções, observa-se que somente a genética consegue melhores resultados, quando se consideram os assuntos divididos em várias categorias (compare-se (a) com (b), ou (c) com (d)), ou quando se consideram diferentes taxas de tráfego entre os assuntos (compare-se (a) com (c)).

Comparação das soluções genéticas obtidas para várias categorias e taxas de tráfego, em por-assunto

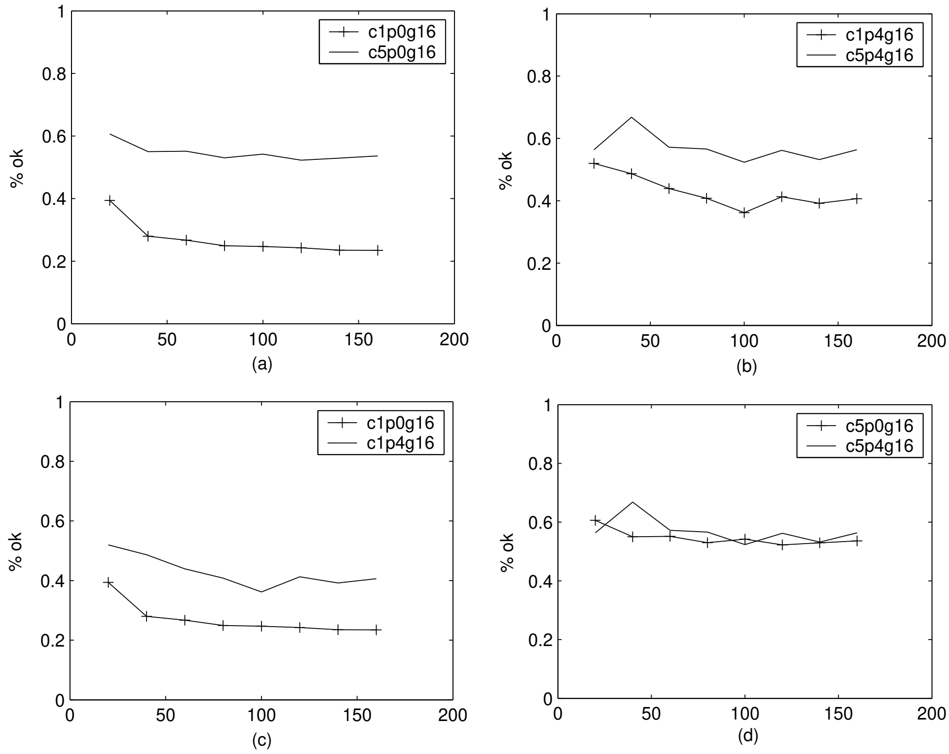


Figura 5.7: Qualidade da solução genética para várias categorias e taxas de tráfego, em por-assunto.

Estes gráficos comparam exclusivamente as soluções genéticas obtidas nos testes atrás, evidenciando o efeito do número de categorias e de diferentes taxas de tráfego. Tanto em (a) como em (b) o aumento do número de categorias de 1 para 5 resultou em melhores emparelhamentos. Isto deve-se ao facto de, quanto maior for o número de categorias para um número total de assuntos constante, maior a probabilidade dos subscritores, que subscrevem assuntos duma mesma categoria, fazerem subscrições semelhantes em volume. Este é o caso de aplicações em que o factor geográfico é determinante na escolha dos assuntos a subscrever.

Os gráficos (c) e (d) mostram também que se considerarmos as taxas de tráfego dos assuntos, melhor é a solução de emparelhamento obtida. Neste caso, o algoritmo genético dá preferência a emparelhamentos que distribuam a carga dos assuntos de maior peso, por diferentes grupos.

Comparação das soluções de difusão total, emparelhamento alternado de assuntos, e genética, em por-conteúdo

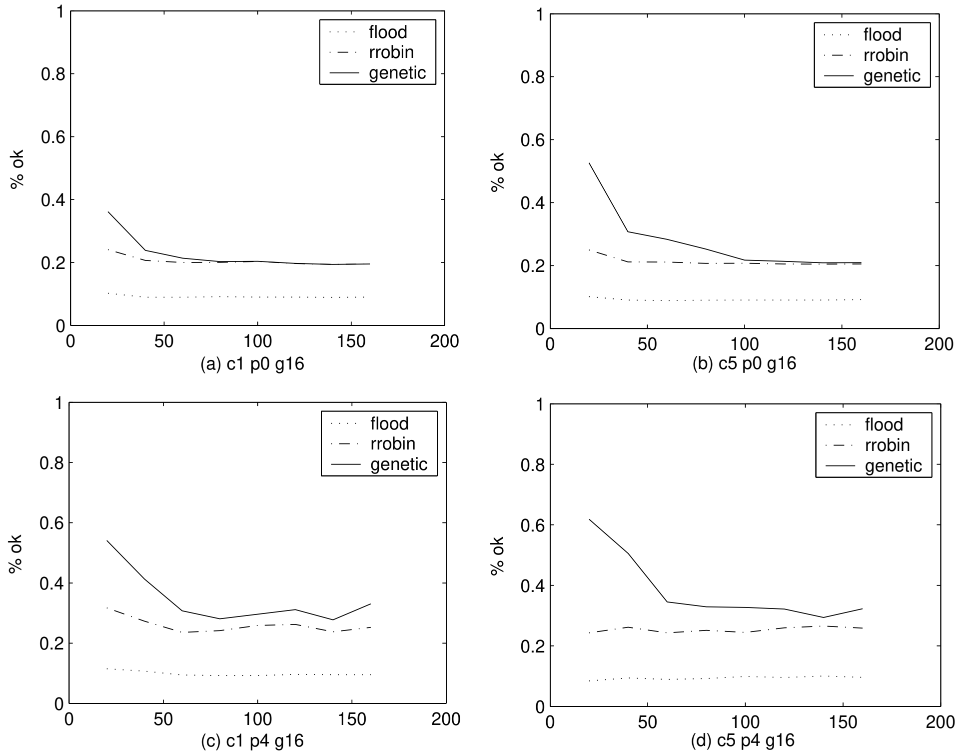


Figura 5.8: Qualidade de diferentes formas de emparelhamento, em por-conteúdo.

Os cenários destes testes foram semelhantes aos realizados nos testes em por-assunto da Fig. 5.6, excepto que em por-conteúdo o raio de subscrição variou entre 10 e 20 volumes de um total de 64 volumes atómicos em cada assunto. Também neste caso, o número de subscrições efectuadas por cada cliente oscilou entre 4 e 8.

Para um número de subscritores superior a ≈ 75, verifica-se que em (a) e (b), apesar do aumento do número de categorias, tanto a solução de emparelhamento alternado como a solução genética obtêm resultados semelhantes. No entanto, considerando em (c) e (d) a taxa de tráfego de cada assunto, a solução genética consegue obter melhores resultados. Comparativamente com os resultados em por-assunto, a maior dificuldade em por-conteúdo reside no facto do número de subscrições diferentes ser muito superior (≈ 450 subscrições para 160 subscritores, em vez de 100 subscrições em por-assunto) para a mesma quantidade de grupos (16).

Comparação das soluções genéticas obtidas para várias categorias e taxas de tráfego, em por-conteúdo

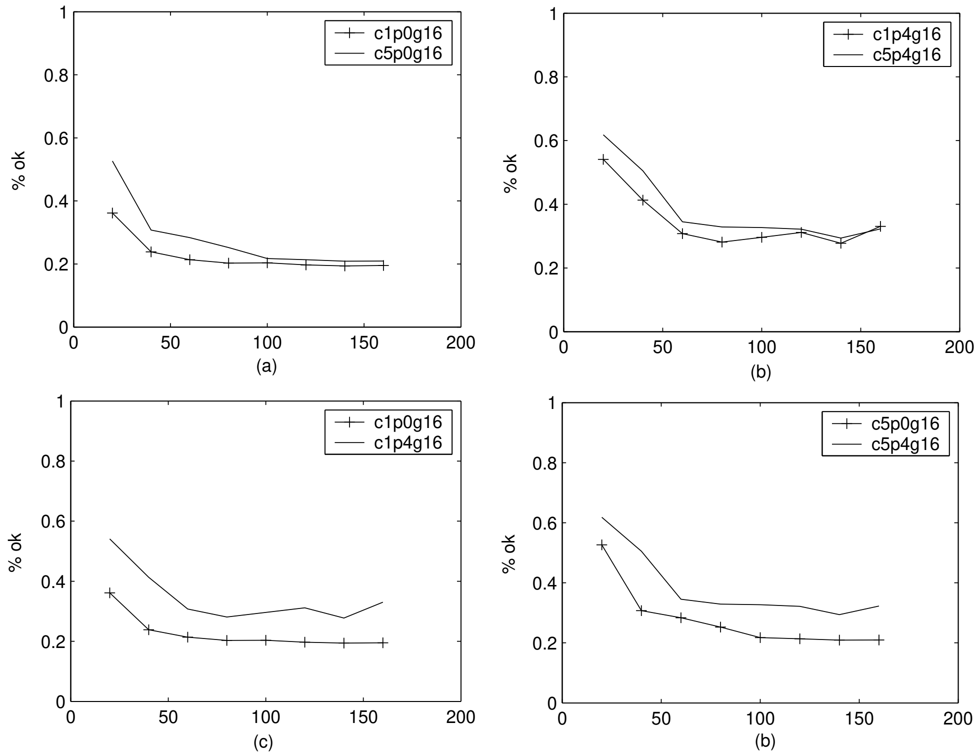
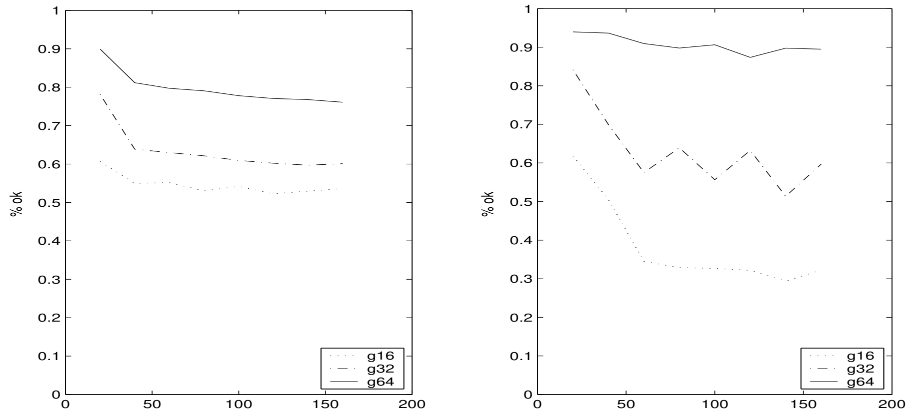


Figura 5.9: Qualidade da solução genética para várias categorias e taxas de tráfego, em por-conteúdo.

Estes gráficos comparam exclusivamente as soluções genéticas obtidas nos testes atrás, evidenciando o efeito do número de categorias e de diferentes taxas de tráfego. As conclusões nesta situação são as mesmas que no caso do por-assunto (Fig. 5.7), embora os melhoramentos nas soluções obtidas apresentem diferenças menos significativas, em virtude de um maior número de subscrições diferentes em por-conteúdo.

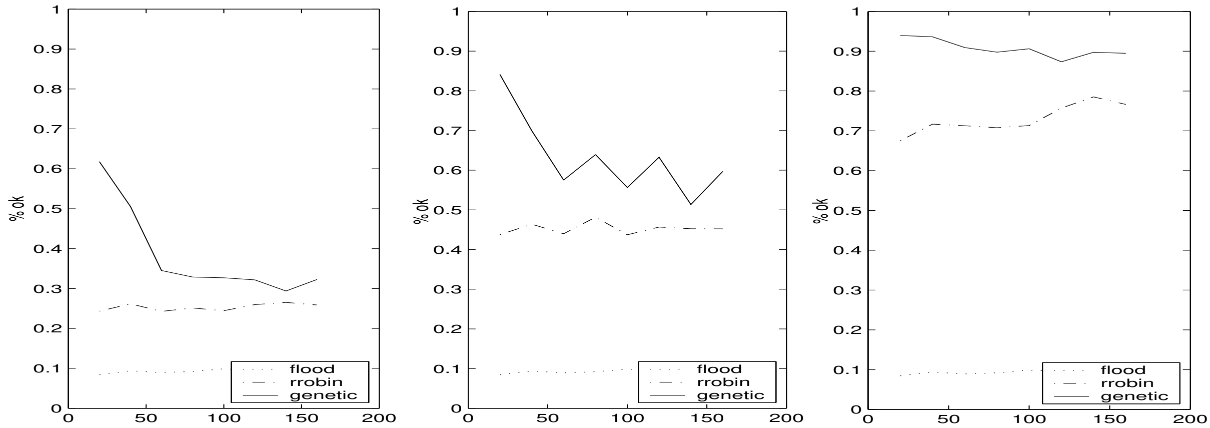
Efeito do aumento do número de grupos de difusão



(a) c5 p0 (b) c5 p4

Figura 5.10: Efeito do aumento do número de grupos de difusão.

Nestes testes, tomaram-se os cenários anteriores para por-assunto e por-conteúdo, e fez-se variar o número de grupos. Como seria de esperar, uma maior disponibilidade de grupos tanto em (a)por-assunto como em (b)por-conteúdo melhora a qualidade das soluções genéticas, reduzindo o número de notificações entregues indevidamente.



(a) c5 p4 g16 (b) c5 p4 g32 (b) c5 p4 g64

Figura 5.11: Qualidade das diferentes formas de emparelhamento para diferentes grupos, em por-conteúdo.

Neste caso, observa-se que o aumento do número de grupos introduz diferenças mais significativas quando se compara a solução genética com a de emparelhamento alternado, em por-conteúdo.

Efeito do aumento do número de subscritores

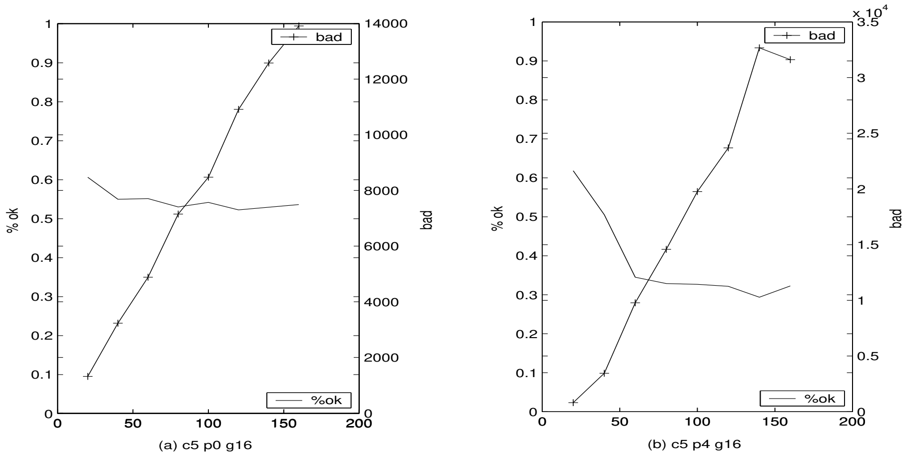


Figura 5.12: Estabilidade da taxa de notificações entregues correctamente.

Tanto em (a)por-assunto como em (b)por-conteúdo, apesar do aumento do número de subscritores, a qualidade das soluções de emparelhamento tende a ser constante quando os restantes parâmetros do cenário se mantém constantes, mesmo que aumente o total de notificações entregues indevidamente. Isto deve-se ao facto de, quanto maior for o número de subscritores, maior é a probabilidade destes fazerem subscrições de volumes semelhantes no espaço de informação.

Efeito do aumento do número de assuntos

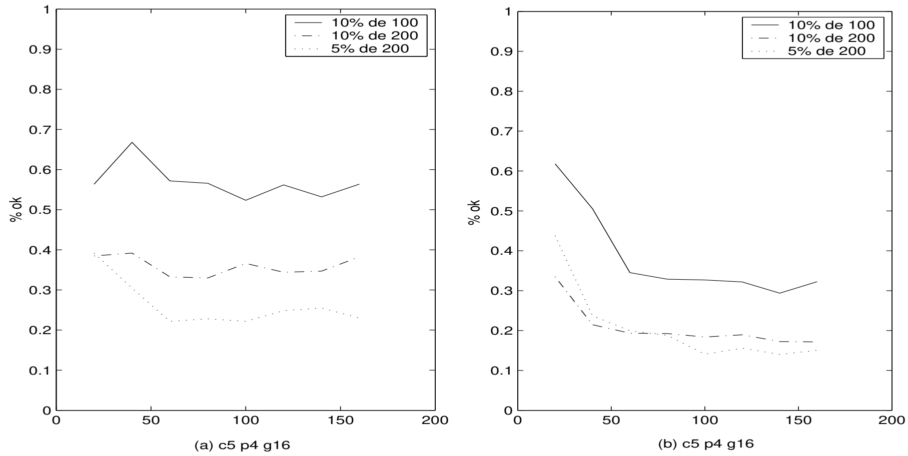


Figura 5.13: Influência do número de assuntos na qualidade das soluções genéticas.

Nestes testes, fez-se variar o número de assuntos subscrevidos por cada subscritor. Verifica-se que com o aumento do número de assuntos, menor o número de subscrições com volumes semelhantes do espaço de informação, quer em (a)por-assunto como em (b)por-conteúdo. Logo, menor a qualidade das soluções obtidas em resultado da menor possibilidade em agrupar subscrições semelhantes.

Outros resultados observados

A selecção por torneio (*tournament selection*) deve ser aplicada entre as $40\% \sim 50\%$ iterações iniciais. A probabilidade de vencer a melhor solução do par de soluções em torneio deverá situar-se entre os $70\% \sim 90\%$. Nas iterações seguintes, o uso da selecção por roleta (*roulette-wheel*) possibilita uma especialização mais rápida da população, convergindo para uma melhor solução.

O cruzamento usando permutação aleatória poderá ser usado até às 5% iterações iniciais, de modo a permitir uma procura mais alargada do espaço de soluções. Nas iterações seguintes, quando se procura aperfeiçoar/convergir uma população de soluções, o mecanismo de permutação não deve ser utilizado.

5.5 Sumário

A simulação do algoritmo de procura genética revelou os factores que mais influenciam a qualidade das soluções obtidas: a organização dos assuntos em categorias; a consideração do peso dos assuntos, ou seja, a sua taxa de tráfego relativa; o tamanho da população de soluções; o uso de elitismo; a probabilidade de cruzamento; e a probabilidade de mutação. Qualquer um destes factores teve impacto na redução do número de notificações entregues indevidamente aos subscritores.

Os resultados sugerem que uma solução de procura genética permite obter melhores emparelhamentos do que uma solução de emparelhamento alternado, nos casos em que as subscrições correspondem a volumes próximos no espaço de informação ou quando se considera as taxas de tráfego relativas dos assuntos. Em por-assunto, o algoritmo genético revelou ser significativamente melhor que a solução alternada, alcançando uma qualidade até três vezes superior (Fig.5.6). Em por-conteúdo (Fig.5.8) verifica-se que as diferenças entre ambas as soluções diminuem com o aumento do número de subscritores. Contudo, neste caso, o aumento do número de grupos faz destacar a solução genética (Fig.5.11).

Estes resultados sugerem que um ponto determinante da qualidade dos emparelhamentos, reside na capacidade de filtrar subscrições de pouco peso ou com um número de subscritores reduzido. Neste sentido, uma possibilidade a explorar consiste em derivar o menor número de subscrições disjuntas em cada assunto, e considerar apenas estas para emparelhamento. A propriedade da disjunção destas subscrições permite que cada uma delas possa ser tratada como se fosse um assunto, simplificando a aplicação de filtros e o cálculo da função de custo.

Por fim, nesta simulação todos os assuntos foram considerados como tendo uma probabilidade de subscrição equivalente. Contudo, certamente existirão aplicações onde isto não acontece, pelo que será revelador simular o comportamento do algoritmo de emparelhamento neste tipo de cenários.

Capítulo 6

Conclusão e Futuros Desenvolvimentos

As soluções actualmente usadas para realizar sistemas de publicação e subscrição de informação possuem formas bastante flexíveis para referenciar a informação. Algumas dessas soluções recorrem à classificação de eventos e incluem a possibilidade de endereçar a informação através das propriedades destes. No entanto, nenhuma delas tem capacidade de escala suficiente para ser utilizada na realização de aplicações genéricas e abertas, na *Internet*.

Neste contexto, este trabalho contribui com uma nova arquitectura para sistemas de publicação e subscrição, assente nos modelos de redes de nós de eventos e de difusão em *IP Multicast*. A conjugação destes dois modelos numa única solução permite reunir a expressividade do primeiro e a capacidade de escala e de evolução do segundo. Contudo, qualquer destes modelos levanta alguns problemas quando aplicados em redes alargadas e abertas. Assim, introduzimos as noções de *domínio de publicação* e de *subscrição-orientada* ao domínio, justificando-as com o seu potencial em aumentar a capacidade de escala destes sistemas nesse tipo de ambientes. Também motivámos para a necessidade de se criar mecanismos eficientes para controlo de acesso aos assuntos em cada domínio, e de assegurar a integridade e autenticidade da informação, em particular quando distribuída em redes abertas. Estes conceitos introduzidos neste trabalho servem de base à criação de um espaço de informação global na *Internet*, centrado em domínios fonte de informação. Como tal, sugerimos uma sintaxe para referenciar a informação neste espaço, baseada no formato *URL*.

O uso da difusão *IP Multicast* para entrega local de notificações levanta alguns desafios, em resultado da reduzida expressividade do seu mecanismo de endereçamento e do risco de degradação do protocolo por utilização excessiva de grupos. Como tal, propusemos e aprofundámos um algoritmo para emparelhamento de subscrições num conjunto limitado de grupos de difusão *IP Multicast*, componente chave da arquitectura apresentada. Este algoritmo executa uma pesquisa genética para ultrapassar a grande dimensão do espaço de soluções e encontrar uma boa solução de emparelhamento. Os testes de simulação realizados revelam que a qualidade deste algoritmo é superior à alcançada por uma solução de difusão total ou de emparelhamento alternado.

De futuro, será interessante desenvolver um protótipo da arquitectura proposta, de modo a estudar o comportamento das várias técnicas e conceitos apresentados, e que atendendo à sua complexidade, não foram possíveis de realizar e avaliar durante o tempo de desenvolvimento desta tese.

Para finalizar, pensamos que a arquitectura e as técnicas apresentadas poderão apoiar o desenvolvimento de aplicações de difusão de informação na *Internet*. Exemplos destas aplicações serão, entre outros, de colaboração remota entre equipas de trabalho, de supervisão e controlo à distância de sistemas, e de distribuição de actualizações de documentos e de software.

Apêndice A

Internet Multicast Allocation Architecture

A difusão em *IP Multicast* [12], usa a Classe D de endereços (na gama 224.0.0.0/4), para comunicação em grupo. Ao contrário das classes de endereços A, B e C para comunicação ponto-a-ponto, nas quais a atribuição de endereços assume um padrão semi-permanente, os endereços de Classe D foram pensados para atribuição dinâmica, sessão-a-sessão, consoante as necessidades das aplicações[1].

Presentemente, uma das questões por resolver consiste no problema de atribuição dinâmica e temporária de endereços de difusão, em toda a *Internet*. A *Internet Multicast Allocation Architecture* surge assim como uma proposta de solução para este problema de atribuição dinâmica de endereços de difusão.

[1]No entanto, alguns endereços Classe D foram permanentemente atribuídos a alguns protocolos especializados ou de suporte a vários mecanismos de rede. São exemplo disso, o endereço 224.0.1.1 usado pelo *Network Time Protocol* e o endereço 224.2.127.255 usado para anunciar sessões de difusão global. Uma lista actualizada dos endereços atribuídos pela *Internet Assigned Numbers Authority* (*IANA*), pode ser consultada em ftp://ftp.isi.edu/in-notes/iana/assignments/multicast-addresses.

A.1 Administrative Scoping

Esta arquitectura assume o conceito de zona administrativa (*administrative scope*) [29], como mecanismo primário de delimitação do tráfego de difusão. Este mecanismo satisfaz a necessidade de circunscrição da propagação de um pacote a uma região da rede, ultrapassando as dificuldades do uso para esse efeito do campo *TTL* (*Time-To-Live*) dos pacotes *IP*. Estas dificuldades devem-se ao facto da parametrização do campo *TTL* se poder tornar complexa e sujeita a falhas, em particular quando se pretende obter várias regiões limite de propagação de pacotes. O campo *TTL* foi especificado como um mecanismo de eliminação de pacotes perdidos na rede (evitando que circulem indefinidamente), sendo decrementado em cada encaminhador (*router*), provocando a eliminação do pacote quando o seu valor atinge zero. No entanto, é muitas vezes utilizado para limitar o alcance de um pacote, sendo inicializado num número correspondente ao caminho máximo que pode percorrer. Contudo, se atendermos ao não determinismo do encaminhamento na rede ou a uma eventual alteração topológica desta, é possível que os pacotes não cheguem a todos os destinatários, ou mesmo que saiam da fronteira pretendida. Verificou-se também que o uso do campo *TTL* provoca problemas na presença de alguns mecanismos de difusão e de corte (*pruning*) em certos protocolos de difusão. Consequentemente, será mais efectiva a definição explícita de uma fronteira de difusão na rede, do que a configuração correcta de valores *TTL*.

Uma zona administrativa estabelece explicitamente uma fronteira topológica para delimitação da propagação de pacotes. Esta fronteira é configurada nos encaminhadores de rede limítrofes, parametrizando nestes um conjunto de endereços de grupo (*multicast scope*), para os quais os pacotes emitidos não passam para fora da zona definida. A *IANA* indicou a gama de endereços 239.0.0.0/8 para definição de zonas administrativas. Estes endereços são atribuídos localmente em cada zona, sendo permitida a sua reutilização em zonas não intersectantes. Nesta gama, os endereços 239.255.0.0/16 estão reservados para a *IPv4 Local Scope*, sendo esta a menor zona administrativa possível, não podendo intersectar com qualquer outra. Se esta gama de endereços não for suficiente, a *IPv4 Local Scope* pode usar as gamas 239.254.0.0/16 e 239.253.0.0/16. A gama 239.192.0.0/14 foi reservada para a *IPv4 Organization Local Scope*, e serve para atribuição de endereços às zonas privadas duma organização, podendo a gama ser extendida nos endereços 239.0.0.0/10, 239.64.0.0/10, e 239.128.0.0/10 se se verificar insuficiente. Adicionalmente, os grupos /24 em todas as zonas administrativas foram reservados para a definição de *grupos relativos*. Estes correspondem a deslocamentos (*offsets*) do grupo com maior endereço dentro da gama da zona. Por exemplo, o deslocamento "0" da *IPv4 Local Scope* equivale ao endereço 239.255.255.255, o deslocamento "1" equivale ao endereço 239.255.255.254, e assim por diante, até ao deslocamento "255" correspondente ao endereço 239.255.255.0. Estes *grupos relativos* foram designados para suportar protocolos de gestão local em todas as zonas administrativas (por exemplo, o SADP [35]). A restante gama de endereços em cada zona, abaixo do espaço /24, está disponível para atribuição dinâmica.

A.2 Multicast-Scope Zone Announcement Protocol

De modo a suportar a gestão de zonas administrativas, foi proposto o protocolo *MZAP* (*Multicast-Scope Zone Announcement Protocol*) [28]. Este protocolo permite a descoberta de zonas administrativas activas em cada ponto da rede, possibilita a atribuição de nomes às zonas, e disponibiliza mecanismos através dos quais é possível detectar configurações erróneas. O *MZAP* é concretizado pelos encaminhadores na fronteira de cada domínio, designados por *ZBRs* (*Zone Boundary Routers*), os quais são configurados com a gama de endereços das zonas de que são fronteira, em uma ou mais das suas interfaces. Cada *ZBR* envia periodicamente mensagens *ZAM* (*Zone Announcement Message*) para o interior de cada zona de que é fronteira, contendo informação sobre a respectiva gama de endereços, o *Zone ID*[2], e os vários nomes da zona. Os *ZBRs* conseguem saber se uma zona não está dentro de outra, e trocam mensagens *NIM* (*Not-Inside Message*) indicando esses relacionamentos. Estas mensagens são enviadas para o endereço de grupo relativo da *Local Scope*, [MZAP-LOCAL-GROUP], e são propagadas por todas as *Local Scopes* no interior da respectiva zona. Qualquer servidor à escuta de mensagens *ZAM* e *NIM* no [MZAP-LOCAL-GROUP], fica a conhecer as zonas existentes bem como as suas relações de intersecção. Existem mais dois tipos de mensagens, *ZCMs* (*Zone Convexity Messages*) e *ZLEs* (*Zone Limit Exceeded*), usadas exclusivamente pelos *ZBRs* para verificar a consistência entre configurações e detectar falhas de configuração.

A.3 Internet Multicast Allocation Architecture

Fundamentada no conceito de zona administrativa, a *IMAA* propõe uma solução de atribuição dinâmica de endereços de difusão, estabelecendo como propriedades importantes, uma grande probabilidade de atribuição bem sucedida de endereços, um tempo de latência reduzido, uma baixa probabilidade de atribuição múltipla de um endereço dentro de um mesmo domínio (*allocation domain*), e uma boa utilização do espaço de endereçamento (evitando excessiva fragmentação).

Esta solução introduz uma arquitectura composta por três níveis protocolares hierárquicos (Fig. A.1). No nível mais baixo da hierarquia (NÍVEL 1), as aplicações de difusão obtêm temporariamente endereços para cada zona administrativa em que se inserem, através de servidores *MAAS* (*Multicast Address Allocation Server*), via utilização de um protocolo, tal como o *MADCAP* (*Multicast Address Dynamic Client Allocation*

[2]Um *ZoneID*, equivale ao par formado pelo menor endereço *IP* dos *ZBR*s da zona definida e pelo menor endereço da respectiva *multicast scope*.

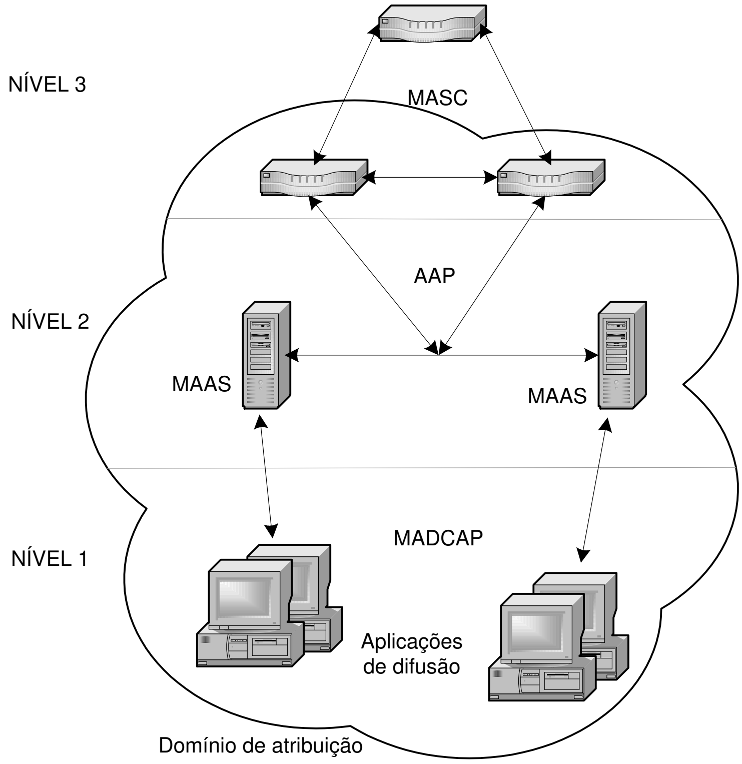


Figura A.1: A *Internet Multicast Allocation Architecture*.

Protocol) [31]. Os *MAAS* aprendem as zonas administrativas e as respectivas gamas de endereços através da captura de mensagens do protocolo *MZAP*. Zonas administrativas de grande dimensão são sub-divididas em domínios de atribuição (*allocation domains*), para eficiência de execução dos protocolos. Nas zonas de pequena dimensão, o domínio de atribuição equivale topologicamente à própria zona.

Quando um *MAAS* atribui um endereço, fica responsável por garantir que o mesmo não é reutilizado no domínio em que é atribuído, durante o respectivo tempo de vida útil. Para tal, os servidores *MAAS* usam um protocolo intra-domínio, o *AAP* (*Address Allocation Protocol*) (NÍVEL 2), para comunicar periodicamente entre si os endereços que reservaram para atribuição às aplicações. O *AAP* funciona em difusão num endereço de grupo relativo à zona administrativa do domínio de atribuição (a "*Allocation Scope*").

Nas zonas administrativas pequenas, os servidores *MAAS* gerem exclusivamente entre si a atribuição da gama de endereços da zona. Por sua vez, em cada domínio duma zona administrativa de grande dimensão, actuam entidades designadas por *Prefix Coordinators*, que realizam um protocolo inter-domínios (NÍVEL 3), para reserva de sub-gamas de endereços da zona. Um protocolo a este nível é o *MASC* (*Multicast Address Set Claim*) [36], onde os *Prefix Coordinators* são usualmente nós encaminhadores.

Os *MAAS* escutam os *Prefix Coordinators* via *AAP*, de forma a obterem as sub-gamas de endereços e respectivos tempos de vida, a atribuir aos clientes dentro dos domínios a que pertencem. Também podem em caso de necessidade, indicar aos *Prefix Coordinators* o seu desejo por mais endereços.

Os *Prefix Coordinators* organizam-se hierarquicamente, e competem entre si pelas gamas de endereços anunciadas por *Prefix Coordinators* hierarquicamente superiores. Toda a comunicação entre estes é realizada via *MASC*. Globalmente, não existe qualquer autoridade central de atribuição de endereços de difusão.

Bibliografia

[1] David Cheriton. Dessimination-oriented communication systems. Technical report, Computer Science Department of Stanford University.

[2] Brian Oki, Manfred Pfluegl, Alex Siegel, and Dale Skeen. The information bus - an architecture for extensible distributed systems. *Operating Systems Review*, 27(5), December 1993.

[3] Douglas E. Comer. *Internetworking With TCP/IP: Principles, Protocols, Architecture*. Prentice Hall, 1991.

[4] Alphorn. Remote procedure calls, user's guide. Technical Report Version 02, The Computer Architecture Group, Asea Brown Boveri Corporate Research, Asea Brown Boveri CH-5405 Baden-Switzerland, February 1989.

[5] H. Bal, M. Kaashoek, and A. Tanenbaum. Implementing distributed algorithms using remote procedure call. In *Proceedings of the National Computer Conference*, pages 499–505. AFIPS, 1987.

[6] TIBCO. Tib/rendezvous white paper. Technical report, TIBCO. Available from http://www.rv.tibco.com/whitepaper.html.

[7] David Glance. Multicast support for data dissemination in orbixtalk. In *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 1996.

[8] Bill Segal and David Arnold. Elvin has left the building: A publish/subscribe notification service with quenching. In *Proceedings of AUUG97*, Brisbane, Australia, September 1997. Available from http://www.dstc.edu.au/.

[9] Antonio Carzaniga. *Architectures for an Event Notification Service Scalable to Wide-area Networks*. PhD thesis, Politecnico di Milano, December 1998. Available from http://www.cs.colorado.edu/˜carnaziga/papers.

[10] P. Mockapetris. Domain names - concepts and facilities. Technical report, IETF, November 1987. RFC 1034.

[11] P. Mockapetris. Domain names - implementation and specification. Technical report, IETF, November 1987. RFC 1035.

[12] S. Deering. Host extensions for ip multicasting. Technical Report RFC 1112, Stanford University, Stanford, CA, USA, August 1989.

[13] Mário Guimarães and Luís Rodrigues. Arquitectura híbrida para publicação e subscrição de informação na internet. In *Actas da 3ª CONFERÊNCIA SOBRE REDES DE COMPUTADORES, CRC'2000*. FCCN, Fundação para a Computação Ciêntífica Nacional, November 2000.

[14] Object Management Group. *Object Management Group. CORBAservices: Common Object Services Specification - Event Service Specification.*, March 1995. Available from http://www.omg.org/library/csindx.html.

[15] S. Maffeis. ibus: The java intranet software bus. Technical report, SoftWired AG, Zurich, Switzerland, February 1997.

[16] IONA. Orbixtalk fact sheet. Technical report, IONA Corporation. Available from http://www.iona.com/products/messaging/talk/index.html.

[17] B. Kantor and P. Lapsley. Network news transfer protocol - a proposed standard for the stream-based transmission of news. Technical report, IETF, February 1986. RFC 977.

[18] G. Cugola, E. Di Nitto, and A.Fuggetta. Exploiting an event-based infrastructure to develop complex distributed systems. In *Proceedings of the 20th International Conference on Software Engineering (ICSE 98)*, Kyoto, Japan, April 1998.

[19] Robert Strom, Guruduth Banavar, Tushar Chandra, Mark Kaplan, Kevan Miller, Bodhi Mukherjee, Daniel Sturman, and Michael Ward. An information flow based approach to message brokering. In *International Symposium on Software Reliability Engineering '98*, 1998. Available from http://www.research.ibm.com/gryphon/.

[20] Hewlett Packard Keryxsoft. *Keryx Version 1.0a Release Notes and Documentation*, 1997. Available from http://keryxsoft.hpl.hp.com/keryx-1.0a/html/index.html.

[21] Object Management Group. *Object Management Group. CORBAservices: Common Object Services Specification - Notification Service Specification.*, March 1998. Available from http://www.omg.org/library/csindx.html.

[22] Sun Microsystems. *Sun Microsystems. Java Message Service.* Available from http://java.sun.com/products/jms.

[23] Patrick Thomas Eugster. *Type-Based Publish/Subscribe*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2001.

[24] Guruduth Banavar, Tushar Chandra, Bodhi Mukherjee, Jay Nagarajarao, Robert E. Strom, and Daniel C. Sturman. An efficient multicast protocol for content-based publish-subscribe systems. In *International Conference on Distributed Computing Systems (ICDCS '99)*, June 1999. Available from http://www.research.ibm.com/gryphon/.

[25] Brian Neil Levine, Jon Crowcroft, Christophe Diot, J. J. Garcia-Luna-Aceves, and James F. Kurose. Consideration of receiver interest for IP multicast delivery. In *INFOCOM (2)*, pages 470–479, 2000.

[26] Arvola Chan. Transactional publish/subscribe: The proactive multicast of database changes. In *SIGMOD'98*, Seattle, WA, USA, 1998.

[27] M. Handley, D. Thaler, and D. Estrin. The internet multicast allocation architecture. Technical report, IETF, October 1999. Internet Draft, draft-ietf-malloc-arch-03.txt.

[28] M. Handley, D. Thaler, and R. Kermode. Multicast-scope zone announcement protocol (mzap). Technical report, IETF, February 2000. RFC 2776.

[29] D. Meyer. Administratively scoped ip multicast. Technical report, IETF, July 1998. RFC 2365.

[30] T. Berners-Lee. Universal resource identifiers in www. Technical report, IETF, June 1994. RFC 1630.

[31] S. Hanna, B. Patel, and M. Shah. Multicast address dynamic client allocation protocol (madcap). Technical report, IETF, December 1999. RFC 2730.

[32] Kevin C. Almeroth. The evolution of multicast: From mbone to inter-domain multicast to internet2 deployment. Technical report, Stardust.com Inc., September 1999. Available from http://www.stardust.com.

[33] Melanie Mitchel. *An Introduction to Genetic Algorithms*. The MIT Press, Cambridge, MA, 1996.

[34] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, 1989.

[35] R. Kermode and D. Thaler. Scoped address discovery protocol (sadp). Technical report, IETF, January 1999. Internet Draft, draft-ietf-mboned-sadp-01.txt.

[36] D. Estrin, R. Govindan, M. Handley, S. Kumar, P. Radoslavov, and D. Thaler. The multicast address set claim (masc) protocol. Technical report, IETF, July 1999. Internet Draft, draft-ietf-malloc-masc-03.txt.

[37] Vicki Johnson and Marjory Johnson. Introduction to ip multicast routing. Technical report, Stardust.com, Inc. Available from http://www.ipmulticast.com/community/whitepapers/introrouting.html.

[38] Lukasz Opyrchal, Mark Astley, Joshua Auerbach, Guruth Banavar, Robert Strom, and Daniel Sturman. Exploiting ip multicast in content-based publish-subscribe systems. Available from http://www.research.ibm.com/gryphon.

[39] M. Handley and S. Hanna. Multicast address allocation protocol (aap). Technical report, IETF, October 1999. Internet Draft, draft-ietf-malloc-aap-02.txt.