Difusão Probabilista com Fiabilidade Semântica

Sérgio Formigo
José Pereira
Luís Rodrigues

DI–FCUL TR–02–10

Agosto 2002

Departamento de Informática
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, 1749–016 Lisboa
Portugal

Technical reports are available at `http://www.di.fc.ul.pt/tech-reports`. The files are stored in PDF, with the report number as filename. Alternatively, reports are available by post from the above address.

Difusão Probabilista com Fiabilidade Semântica*

Sérgio Formigo
Universidade de Lisboa
formigo@di.fc.ul.pt

José Pereira
Universidade do Minho
jop@di.uminho.pt

Luís Rodrigues
Universidade de Lisboa
ler@di.fc.ul.pt

Agosto 2002

Resumo

Os protocolos probabilistas, também designados por protocolos de propagação epidémica, têm recebido recentemente um interesse crescente devido à sua capacidade de escala. Uma das principais limitações deste tipo de protocolos é que podem consumir uma excessiva largura de banda. Este artigo discute um protocolo que combina a difusão probabilista com a fiabilidade semântica. Este novo protocolo alia o suporte de um elevado número de participantes com a tolerância de receptores lentos e congestão na rede. Nomeadamente, é apresentada a concretização do protocolo usando a moldura de objectos de suporte à composição e execução de micro-protocolos *Appia* assim como avaliação do seu desempenho.

Palavras Chave: Difusão probabilista, grande escala, modularidade.

1 Introdução

Em diversas aplicações distribuídas existe a necessidade de difundir informação de modo fiável para um elevado número de destinatários. Protocolos de difusão fiável deterministas, como o RMTP [13] ou suportando sincronia virtual [2] não possuem a capacidade de escala que permita realizar difusão fiável para grupos de grande dimensão (centenas de milhar).

Os protocolos probabilistas, também designados por protocolos de propagação epidémica, têm recebido recentemente um interesse crescente, devido à sua capacidade de escala. Como o seu nome sugere, este tipo de protocolos reproduz o

*Este trabalho foi parcialmente suportado pelo projecto RUMOR (POSI/40088/CHS/2001) e pela Microsoft Research.

método de propagação de epidemias na natureza: um membro "infectado" reencaminha uma mensagem para um sub-conjunto aleatório de membros do grupo. Cada participante repete este processo levando a que, com elevada probabilidade, todos os destinatários acabem por receber a mensagem.

Uma desvantagem deste tipo de protocolos é que podem originar um elevado número de mensagens e, eventualmente, causar a congestão da rede. Este artigo explora uma ideia que consiste em combinar protocolos probabilistas com a noção de *fiabilidade semântica* [14]. A fiabilidade semântica é um modelo de coerência que explora conhecimento acerca do conteúdo das mensagens para definir relações de obsolescência entre estas. A noção de fiabilidade semântica fornece aos protocolos probabilistas um mecanismo para descartar mensagem obsoletas. Estas ideias foram inicialmente propostas em [16] onde é feita uma avaliação preliminar desta arquitectura recorrendo a simulação.

Este artigo apresenta a concretização desta arquitectura na linguagem Java recorrendo a uma moldura de objectos de suporte à composição e execução de micro-protocolos, denominada *Appia* [12]. Esta concretização permite validar os resultados obtidos previamente com simulação através de testes num sistema real. Por outro lado, esta concretização ilustra também as vantagens da concretização modular, uma vez que grande parte da avaliação foi obtida desenvolvendo camadas de teste e registo de informação que podem ser activadas ou removidas de forma transparente para as camadas que concretizam o protocolo.

O resto do texto está organizado da seguinte forma: A Secção 2 motiva o trabalho, apresenta de forma resumida os conceitos de difusão probabilista, fiabilidade semântica e de sistemas editor-subscritor. A Secção 3 introduz a arquitectura do protocolo combinado. A Secção 4 discute a concretização do protocolo no âmbito do sistema de composição de micro-protocolos *Appia*. A Secção 5 apresenta os resultados experimentais obtidos e a Secção 6 conclui o artigo.

2 Motivação

Existem diversas aplicações que necessitam de realizar a difusão fiável de informação para um elevado número de destinatários exigindo ao mesmo tempo um elevado débito. No entanto, a prática tem revelado que protocolos deterministas não conseguem satisfazer estes requisitos [17, 2]. A obtenção de uma garantia determinista de entrega obriga a que as mensagens sejam armazenadas até que seja confirmada a sua recepção. Mesmo com um mecanismo eficiente de confirmação [7], basta que um destinatário consuma mais lentamente as mensagens ou que uma secção da rede esteja congestionada para que exista uma degradação global do desempenho: O espaço disponível em *buffers* no caminho para o ponto de congestão esgota-se e os emissores são impedidos de enviar novas

mensagens, afectando assim todos os receptores. Duas propostas para ultrapassar este problema, conhecido como o "bebé chorão" [18], são a difusão probabilista [3] e a fiabilidade semântica [14].

2.1 Difusão Probabilista

Num protocolo de difusão probabilista cada participante reenvia cada mensagem recebida para um subconjunto aleatório de outros participantes, sendo cada mensagem reenviada um número limitado de vezes e depois eliminada. Apesar da sua simplicidade, esta estratégia assegura uma garantia probabilista de fiabilidade [3]. Este processo de difusão é descrito por equações semelhantes às usadas para descrever o contágio em epidemiologia, sendo por isso conhecidos também como protocolos de propagação epidémica [10]. Variantes deste protocolo [6, 11] diferem quanto aos mecanismos usados para limitar as retransmissões, quanto à topologia da rede assumida e quanto manutenção da composição do grupo.

Estes protocolos são interessantes do ponto de vista da fiabilidade, do funcionamento em grande escala e do desempenho. As garantias de fiabilidade resultam de uma distribuição de probabilidade bimodal: mesmo em presença de falhas de participantes e perdas de mensagens na rede, a probabilidade de uma mensagem ser recebida por todos ou nenhum dos destinatário é elevada relativamente à probabilidade de uma mensagem ser recebida por alguns, mas não todos os destinatários. Ao evitarem a dependência de qualquer componente centralizado para encaminhar e retransmitir mensagens, bem como de qualquer mecanismo de realimentação para confirmar a recepção, estes protocolos podem ser utilizados para difusão de mensagens para um grande número de destinatários. O desempenho destes protocolos é também imune a destinatários mais lentos, que embora possam não receber todas as mensagens não perturbam o resto do grupo.

As garantias de fiabilidade dependem porém do pressuposto que a perda de mensagens na rede está uniformemente distribuída tanto no tempo como relativamente à origem e destino. No entanto, este pressuposto não é válido em situações de congestão da rede e sobrecarga dos participantes. Nestas situações, a memória disponível nos componentes em sobrecarga esgota-se, originando perdas sucessivas de mensagens. Além disso, os protocolos de difusão probabilista conhecidos não dispõem de mecanismos de controlo de congestão. Esta limitação é especialmente grave na medida em que a largura da banda consumida por este tipo de protocolos pode ser considerável. Esta situação torna particularmente difícil a sua utilização em redes em grande escala como a Internet [9].

2.2 Fiabilidade Semântica

Independentemente do protocolo de difusão utilizado, em diversas aplicações existem mensagens que ficam obsoletas pouco depois de serem enviadas, ou seja, a correcção da aplicação não é afectada se algumas mensagens (seleccionadas) se perderem. Isto acontece porque o seu conteúdo se torna irrelevante ou está implícito em outras mensagens enviadas pouco tempo depois. Por exemplo, numa aplicação que difunde o valor actualizado de um conjunto de items de dados, cada mensagem torna obsoletas as versões anteriores do mesmo item.

Em situações de congestão, em que as mensagens se acumulam nos *buffers*, podem encontrar-se mensagens enviadas com algum tempo de intervalo e é possível identificar mensagens obsoletas ainda em trânsito. Estas mensagens podem ser imediatamente eliminadas, o que permite que:

- o espaço libertado pode ser usado para novas mensagens fazendo com que o emissor não seja bloqueado por controlo de fluxo, ou pelo menos, que seja bloqueado por menos tempo e com menos frequência;
- os destinatários mais lentos não necessitem de processar mensagens obsoletas e possam permanecer no grupo sem perturbar o seu desempenho global.

O protocolo resultante é fiável apenas para as mensagens que não se tornam obsoletas, pelo que considera a semântica das mensagens resultando num critério de *fiabilidade semântica*.

Para tirar partido da fiabilidade semântica é necessário disponibilizar um mecanismo que permita ao protocolo identificar mensagens relacionadas e determinar quais as obsoletas. A quantidade de mensagens que pode ser eliminada depende da configuração e estado do sistema, em particular, do tamanho e utilização média dos *buffers* [14] que determinam a distância máxima entre mensagens relacionadas das quais uma pode ser identificada como obsoleta. Num sistema não congestionado, em que a utilização média dos *buffers* é baixa, não é observável a eliminação de mensagens. Esta é máxima em situações de congestão em que os *buffers* se encontram cheios.

A eficácia do protocolo é também, logicamente, afectada pelas características do tráfego gerado pela aplicação. A replicação de um servidor para jogos multi-utilizador permitiu observar mais de 40% de mensagens obsoletas, o que se traduz na possibilidade de acomodar participantes até 40% mais lentos [15].

2.3 Aplicações Alvo

O paradigma de comunicação editor-subscritor [5] tem vindo a ganhar uma aceitação crescente no desenvolvimento de aplicações distribuídas. A interacção entre

produtores e consumidores da informação é feita indirectamente: Cada subscritor regista junto do sistema os seus interesses. Cada vez que um editor publica informação, ela é encaminhada pelo sistema para os subscritores interessados. Deste modo é possível desenvolver a aplicação independentemente do número e localização quer de produtores quer de consumidores de informação.

Quando existe um número elevado de subscritores interessados num dado tópico e estes se encontram uniformemente distribuídos pela rede, a solução mais eficiente para a concretização de um sistema editor-subscritor é o recurso a protocolos de difusão fiável [1]. Num sistema de utilização geral, este mecanismo deve pois estar disponível como complemento ao encaminhamento baseado no conteúdo [4] usado quando o interesse num tópico é localizado.

Uma aplicação de sistemas editor-subscritor são os sistemas de leilões distribuídos: O valor actual da licitação de cada item é publicado pelo serviço de leilões. Cada licitante poderá subscrever a informação sobre os itens em que está interessado, possivelmente publicada por diferentes servidores. Neste caso, cada novo valor publicado torna os anteriores valores para o mesmo item obsoletos.

Outra aplicação são os jogos distribuídos na Internet. Normalmente, o estado do jogo consiste num conjunto de entidades com um conjunto de propriedades. Por exemplo, num sistema de simulação de voo, cada entidade é um avião cujas propriedades são a posição e velocidade no espaço tri-dimensional. Cada jogador deverá então subscrever a informação dos aviões que se encontram no seu espaço visual. Um sistema de controlo de tráfego aéreo deve subscrever a informação de todas as entidades. Mais uma vez, cada nova versão das propriedades de uma entidade torna a versão anterior obsoleta.

Note-se que em ambas as aplicações, não pode ser previsto que uma mensagem se vai tornar obsoleta. Se uma licitação dá origem à transacção final, o seu valor deve ser recebido de modo fiável por todos os interessados. No caso do simulador de voo, se um movimento dá origem a uma colisão, deve também ser conhecido por todos os observadores. Isto faz com que a utilização de um protocolo de difusão não fiável seja insuficiente. Torna-se então necessário utilizar um protocolo de difusão fiável.

Por um lado, as características de um protocolo probabilista são desejáveis para assegurar que o sistema é adequado a situações em que existe um elevado número de subscritores. Por outro lado, na presença de receptores mais lentos, a perda selectiva de mensagens proporcionada pela fiabilidade semântica permite acomodar melhor receptores com recursos limitados. Torna-se pois interessante a combinação de ambas as propostas num único protocolo.

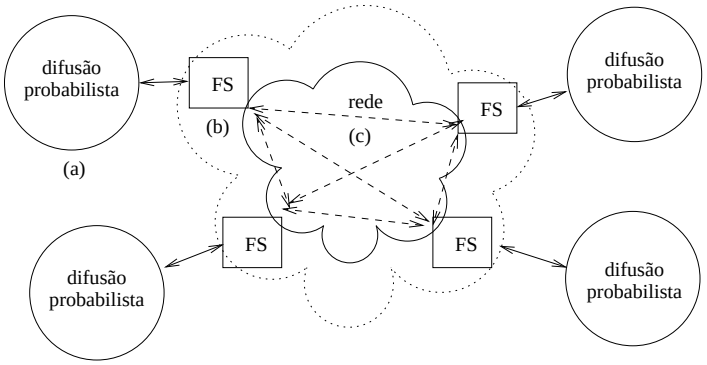

Figura 1: Arquitectura do protocolo: (a) difusão probabilista; (b) fiabilidade semântica; e (c) rede.

3 Difusão Probabilista com Fiabilidade Semântica

Uma combinação simplista de difusão probabilista com fiabilidade semântica pode ser obtida tomando um protocolo que faça explicitamente o armazenamento temporário de mensagens durante o processo de reenvio [6] e identifique e elimine mensagens obsoletas nesse *buffer*. Esta abordagem apresenta dois inconvenientes importantes [16]:

- não relaciona o tamanho dos *buffers* ocupados com a congestão da rede, pelo que a eliminação de mensagens não é regulada automaticamente, nomeadamente, é independente da congestão do sistema e só pode ser alterada modificando os parâmetros do protocolo probabilista;
- não efectua qualquer controlo de congestão na utilização que faz da rede, o que impossibilita a sua utilização em redes partilhadas e não evita perdas significativas de mensagens que podem comprometer as garantias de fiabilidade oferecidas pelo protocolo.

Como consequência, é difícil obter uma configuração do protocolo de difusão óptima tanto do ponto de vista probabilista como da identificação e eliminação de mensagens obsoletas [16].

3.1 Arquitectura

Para ultrapassar esta limitação introduzimos uma arquitectura em três camadas que separa as questões ligadas à fiabilidade semântica da difusão probabilista e do controlo de congestão da rede, esquematizada na Figura 1.

Camada de acesso à rede A camada mais baixa da arquitectura proposta para o protocolo é responsável pelo envio de mensagens ponto-a-ponto. As ligações ponto-a-ponto utilizadas não necessitam de ser fiáveis. No entanto, é necessário que essas ligações não sejam susceptíveis de perdas sucessivas na mesma ligação ou correlacionadas entre diversas ligações, por exemplo em situação de congestão. A utilização de um protocolo de datagramas não fiável, como UDP/IP, está pois fora de questão. Por outro lado, esta camada deve minimizar a quantidade de *buffers* usada. Sempre que necessário, nomeadamente quando a largura de banda disponível é insuficiente devido a um pico de carga, o armazenamento deve ser feito na camada seguinte.

Camada de fiabilidade semântica A camada de fiabilidade semântica efectua o armazenamento das mensagens que vão ser enviadas. Esta camada funciona sobre a camada de acesso à rede, a qual pode bloquear em caso de congestão. Mesmo que tal aconteça, a camada de fiabilidade semântica deve continuar a aceitar mensagens da camada superior, o protocolo probabilista. Admitindo que o espaço de armazenamento disponível é limitado, só é possível se em caso de sobrecarga forem eliminadas mensagens. As mensagens a eliminar podem ser escolhidas de duas maneiras:

- Identificando mensagens obsoletas: Se uma mensagem pendente para um receptor é tornada obsoleta por qualquer outra mensagem também para o mesmo receptor, pode ser eliminada. Este comportamento impede que mensagens sejam descartadas em excesso aumentado a quantidade de informação que é entregue aos receptores mais lentos.

- Aleatoriamente: Se existem no *buffer* demasiadas mensagens destinadas a um dado receptor, é escolhida aleatoriamente uma das mensagens destinada a esse receptor para ser eliminada.

A preocupação fundamental na concretização desta camada é organizar o armazenamento por forma a que ambas as operações de eliminação de mensagens sejam eficientes.

Camada de difusão probabilista A camada de difusão probabilista determina a politica utilizada para reencaminhar cada mensagem conhecida, tanto quanto à escolha dos destinatários como quanto à limitação das retransmissões de cada mensagem. Nesta camada pode ser utilizado qualquer dos algoritmos existentes [3, 6].

3.2 Discussão

O funcionamento da arquitectura proposta está intimamente dependente dos pressupostos necessários para a difusão probabilista. Mais precisamente, sabe-se que estes protocolos oferecem garantias de fiabilidade mesmo com elevadas taxas de perdas, desde que estas perdas sejam uniformemente distribuídas [10]. A nossa aproximação tenta assegurar que estes pressupostos são válidos para as mensagens relevantes, descartando sobretudo mensagens obsoletas. Com isso libertam-se recursos, nomeadamente *buffers* e largura de banda, que pode ser utilizada para minimizar as perdas de mensagens ainda não obsoletas. Além disso, as perdas nas mensagens não obsoletas deixam de acontecer na rede, onde podem ser correlacionadas, mas sim nos *buffers* onde se pode controlar o processo de descarte para aproximar a uniformidade desejada. Adicionalmente, o protocolo passa a fazer controlo de congestão, pelo que pode ser utilizado em redes partilhadas como a Internet.

4 Concretização

Os protocolos foram concretizados na linguagem Java, utilizando o sistema *Appia* [12] na construção da arquitectura para uma maior facilidade na sua concretização.

4.1 Appia

O *Appia* [12] é um sistema modular de suporte à comunicação. Cada módulo do *Appia* é uma camada, i.e. um micro-protocolo responsável por garantir uma determinada propriedade. Estas camadas são independentes e podem ser combinadas. Essa combinação constitui uma pilha de protocolos que oferece uma Qualidade de Serviço (QdS) com as propriedades desejadas.

Definida uma QdS é possível criar um ou mais *canais* de comunicação. A cada canal está associado uma pilha de *sessões*, uma para cada camada de protocolo. O objecto sessão permite manter o estado necessário à execução do protocolo da camada correspondente. Por exemplo, uma camada concretizando ordenação FIFO necessita de manter um ou mais números de sequência como parte do seu estado. O *Appia* permite que canais diferentes partilhem sessões se assim o desejarem. Usando uma vez mais o exemplo de uma camada FIFO, é possível criar vários canais em que todas as mensagens partilham o mesmo número de sequência (através da partilha da sessão correspondente) ou canais cujas mensagens são ordenadas de modo independente (usando uma sessão diferente do protocolo FIFO para cada canal).

A interacção entre camadas é realizada através da troca de eventos. Os eventos são tipificados e cada camada declara ao sistema quais os eventos que cria e que está interessada em processar. O sistema optimiza o fluxo de eventos na pilha de protocolos, assegurando que os eventos só são entregues às camadas que registaram interesse no seu processamento.

4.2 Camadas Concretizadas

A concretização da arquitectura proposta sobre *Appia* recorre também a três camadas. A comunicação entre elas processa-se através do evento `OSendableEvent`, que corresponde a uma mensagem que pode ser enviada na rede. Isto torna possível a utilização de cada uma destas camadas independentemente e em conjunto com camadas existentes no sistema *Appia*. De forma a permitir a identificação e eliminação de mensagens obsoletas, um `OSendableEvent` pode ser anotado pela aplicação com um tipo e um número de versão. É considerada obsoleta uma mensagem se existir outra do mesmo tipo, enviada pelo mesmo emissor e cuja número de versão seja superior.

Camada de difusão probabilista Esta camada é responsável pela transmissão das mensagens bem como pela manutenção da composição do grupo de uma forma também probabilista.

Dada a natureza epidémica do processo de disseminação, é possível que a mesma mensagem seja recebida várias vezes pelo mesmo participante. No entanto, apenas a primeira cópia da mensagem deve ser entregue à aplicação. A utilização de números sequência para identificar duplicados não é conveniente pois isso obrigaria cada nó a manter estado para cada um dos emissores. A solução passa pois por guardar a identificação de cada mensagem recebida numa tabela de *hashing* durante um período de tempo suficiente para que o numero máximo de retransmissões aconteça [6].

A retransmissão de uma mensagem recebida é efectuada apenas no caso de o número máximo de retransmissões nessa mensagem não ter sido atingido e consiste em [8]:

- incrementar no campo correspondente no cabeçalho da mensagem o número de retransmissões efectuadas;
- escolher aleatoriamente alguns dos destinatários conhecidos;
- notificar a camada inferior para fazer o envio dessa mensagem para cada um dos destinatários escolhidos.

A manutenção da composição do grupo é feita usando o algoritmo proposto em [6]. Este algoritmo acrescenta informação de controlo às mensagens de dados que permite manter, em cada participante, uma *vista* parcial da filiação global no grupo. O objectivo de se manter uma vista parcial é o de assegurar a capacidade de escala do protocolo, reduzindo o espaço necessário para manter informação de filiação. Mesmo assim, o protocolo assegura que em cada participante é mantida a informação necessária para assegurar a fiabilidade da difusão. O modo de funcionamento deste serviço é o seguinte:

- cada participante mantém um lista de membros conhecidos do grupo, uma lista de membros que se juntaram recentemente ao grupo e uma lista de membros que abandonaram recentemente o grupo;
- no envio de cada mensagem de dados, é acrescentada a lista dos membros que se juntaram (abandonaram) recentemente o grupo;
- após a recepção de uma mensagem de dados, a listas são actualizadas com a informação recebida.

Se o tamanho de qualquer das listas ultrapassa um limite pré-estabelecido, os elementos são removidos aleatoriamente. Isto assegura que a informação acerca da filiação no grupo é distribuída de forma uniforme pelas diversas vistas parciais.

Camada de fiabilidade semântica A camada de fiabilidade semântica mantém um *buffer* para cada destino solicitado pela camada superior. A gestão desta camada é dinâmica, mantendo informação apenas sobre os destinos para os quais existem mensagens pendentes, adaptando-se assim à gestão do grupo efectuada pela camada superior. Uma das características fundamentais é a gestão de memória. Sempre que uma mensagem é enviada para vários destinos, é guardada apenas uma cópia da mensagem indexada nos vários *buffers*.

De forma a acelerar o processo de detecção e eliminação de mensagens obsoletas, as mensagens presentes nesta camada são também indexadas segundo o seu tipo e número de versão. Desta forma é possível imediatamente durante a chegada de uma mensagem saber quais estão obsoletas e que podem ser removidas.

Camada de acesso à rede As alternativas consideradas para camada de acesso à rede são:

- concretização de uma camada que disponibilize uma ligação ponto-a-ponto não fiável mas com controlo de congestão adequado à utilização na Internet.
- utilização de ligações TCP/IP.

A solução adoptada foi a utilização de ligações TCP/IP. Esta opção obrigou o desenvolvimento de uma nova camada do *Appia*, a `TCPControl`. Esta camada extende a funcionalidade da camada `TCPComplete` do *Appia*, a qual oferece uma robustez e desempenho já testados e é adequada à utilização na Internet. A nova funcionalidade concretiza um protocolo local de controlo de fluxo com a camada de fiabilidade semântica, de modo a assegurar que as mensagens são armazenadas por esta última camada e apenas são processadas pela camada `TCP-Control` quando é possível enviar dados para o canal TCP correspondente. Por omissão, utiliza-se uma quantidade mínima de *buffers* em ambos os extremos de uma ligação TCP/IP. Isto faz, que a ocupação dos *buffers* da camada de fiabilidade semântica seja superior, aumentando as oportunidades de identificação e eliminação de mensagens obsoletas.

4.3 Vantagens do Sistema Appia

O *Appia*, ao fornecer suporte à composição de protocolos facilitou a concretização da arquitectura modular descrita nos parágrafos anteriores. Cada camada é independente do modo como cada uma das restantes camadas está concretizada, sendo apenas dependente dos eventos trocados. Estes eventos representam a transmissão de mensagens e o controlo de fluxo local entre a camada de acesso à rede e a camada de fiabilidade semântica. Durante o processo de desenvolvimento, em que estas camadas foram sucessivamente depuradas, esta modularidade facilitou a comparação entre versões distintas da mesma camada. Outra vantagem do *Appia* é que os mecanismos de composição facilitam a inserção de camadas de controlo e teste para realizar a avaliação de desempenho do protocolo.

5 Avaliação do Desempenho

O objectivo do desenvolvimento do protocolo de difusão probabilista com fiabilidade semântica pode ser resumido como pretendendo garantir a entrega das mensagens que nunca ficam obsoletas. Como tal o critério de avaliação utilizado é o número de mensagens recebidas por todos os participantes. Mais precisamente, interessa saber se as mensagens que num dado padrão de tráfego nunca ficam obsoletas são recebidas atomicamente.

Para o efeito, foram utilizadas 13 estações de trabalho Pentium II a 300MHz, com o sistema operativo RedHat Linux 7.2, a máquina virtual IBM Java 1.4.0 e ligadas por uma rede de 10 Mbits. Além da pilha protocolo de difusão em estudo, em cada um dos participantes foi configurada uma segunda pilha para controlo dos testes através de ligações TCP/IP. O inicio dos testes é efectuado através da pilha de controlo a partir de uma das estação de trabalho.

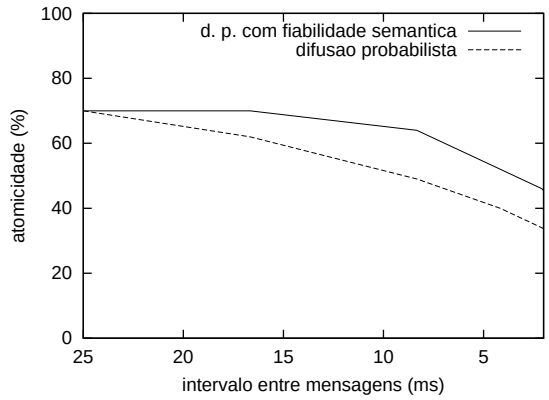

Figura 2: Atomicidade de protocolos de difusão probabilista (percentagem de mensagens recebida por todos os participantes).

Cada teste efectuado consistiu no envio periódico de mensagens por cada um dos 12 participantes. O tráfego é gerado com um padrão de obsolescência em que 50% das mensagens nunca ficam obsoletas e as restantes mensagens são repartidas por diversos tipos. Este tipo de tráfego corresponde ao que é encontrado em aplicações observadas, nomeadamente, em jogos multi-utilizador [15].

Foram realizados vários testes, com um intervalo entre mensagens cada vez mais pequeno. A identificação das mensagens recebidas por cada um dos nós é registada localmente e mais tarde processada para obter a medida de atomicidade. O resultado é apresentado na Figura 2. Observa-se que, sem atender à obsolescência, a partir de 25 ms, correspondendo a 40 msg/s no sistema, a atomicidade começa a degradar-se. Por exemplo, com 8 ms apenas 49% das mensagens são recebidas por todos os participantes. Isto acontece devido às limitações do tempo de processamento disponível para cada mensagem. Os testes com o protocolo com fiabilidade semântica mostram que na mesma situação 64% das mensagens que nunca ficam obsoletas são ainda entregues atomicamente.

6 Conclusão

Este artigo descreveu a concretização e avaliação de um protocolo de difusão probabilista que suporta um modelo de fiabilidade semântica. O protocolo preserva as qualidades de escala e robustez dos protocolos epidémicos, pois não acrescenta nenhum mecanismo que dependa do grupo como um todo. Ao mesmo tempo evita a perda correlacionada de mensagens na rede devido a fenómenos de congestão através do descarte selectivo das mensagens obsoletas.

O protocolo foi concretizado através da composição de três micro-protocolos: uma camada de acesso à rede, uma camada de fiabilidade semântica e uma camada de difusão probabilista. Para realizar esta concretização recorreu-se ao sistema *Appia*. Este sistema, ao fornecer suporte explicito para a composição de protocolos, facilitou as experiências de avaliação, uma vez que permitiu acrescentar de modo transparente várias camadas de teste, como por exemplo as camadas que registam a transmissão e recepção de mensagens para posterior tratamento estatístico.

O testes efectuados ao protocolo revelam que a utilização de fiabilidade semântica permite de facto melhorar as garantias de difusão atómica das mensagens que nunca ficam obsoletas. Isto faz com que seja possivel a uma aplicação operar correctamente em situações de carga em que o protocolo original provocaria incoerência devido à incapacidade de assegurar a atomicidade.

Referências

[1] G. Banavar, T. Chandra, B. Mukherjee, R. Strom J. Nagarajarao, and D. Sturman. An efficient multicast protocol for content-based publish-subscribe systems. In *The 19th IEEE International Conference on Distributed Computing Systems (ICDCS '99)*, 1999.

[2] K. Birman. A review of experiences with reliable multicast. *Software Practice and Experience*, 29(9):741–774, Junlo 1999.

[3] K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky. Bimodal multicast. *ACM Transactions on Computer Systems*, 17(2):41–88, Maio 1999.

[4] A. Carzaniga, D. Rosenblum, and A. Wolf. Achieving scalability and expressiveness in an Internet-scale event notification service. In *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing (PODC-00)*, pages 219–228, NY, Junlo 16–19 2000. ACM Press.

[5] P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. Technical Report DSC ID:2000104, EPF Lausanne, 2001.

[6] P. Eugster, R. Guerraoui, S. Handrukande, A.-M. Kermarrec, and P. Kouznetsov. Lightweight probabilistic broadcast. In *Intl. Conf. on Dependable Systems and Networks (DSN'2001)*, 2001.

[7] K. Guo. *Scalable Message Stability Detection Protocols*. PhD thesis, Cornell Univ., Computer Science, Maio 1998.

[8] M. Hayden and K. Birman. Probabilistic broadcast. Technical Report TR96-1606, Cornell University, Computer Science, 1996.

[9] V. Jacobson. Congestion avoidance and control. *ACM Computer Communication Review; Proceedings of the Sigcomm '88 Symposium in Stanford, CA, August, 1988*, 18(4):314–329, 1988.

[10] A.-M. Kermarrec, L. Masssoulié, and A. Ganesh. Reliable probabilistic communication in large-scale information dissemination systems. Technical Report 2000-105, Microsoft Research, 2000.

[11] M.-J. Lin and K. Marzullo. Directional gossip: Gossip in a wide area network. Technical Report CS1999-0622, University of California, San Diego, Computer Science and Engineering, Junho 16, 1999.

[12] H. Miranda, A. Pinto, and L. Rodrigues. Appia, a flexible protocol kernel supporting multiple coordinated channels. In *Proceedings of the 21st International Conference on Distributed Computing Systems*, pages 707–710, Phoenix, Arizona, Abril 2001. IEEE.

[13] S. Paul, K. Sabnani, J. Lin, and S. Bhattacharyya. Reliable multicast transport protocol (rmtp). *IEEE Journal on Selected Areas in Communications*, 15(3):407–421, Abril 1997.

[14] J. Pereira, L. Rodrigues, and R. Oliveira. Semantically reliable multicast protocols. In *Proceedings of the Nineteenth IEEE Symposium on Reliable Distributed Systems*, pages 60–69, Outubro 2000.

[15] J. Pereira, L. Rodrigues, and R. Oliveira. Reducing the cost of group communication with semantic view synchrony. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, page (to appear), Washington (DC), USA, Junho 2002.

[16] J. Pereira, L. Rodrigues, R. Oliveira, and A.-M. Kermarrec. Probabilistic semantically reliable multicast. Technical report, Department of Computer Science, University of Lisbon., Agosto 2001.

[17] R. Piantoni and C. Stancescu. Implementing the Swiss Exchange Trading System. In *Proceedings of The Twenty-Seventh Annual International Symposium on Fault-Tolerant Computing (FTCS'97)*, pages 309–313. IEEE, Junho 1997.

[18] B. Quinn and K. Almeroth. RFC 3170: IP Multicast applications: Challenges and solutions. IETF Network Working Group, Setembro 2001.