Simulação de um Protocolo de Descoberta de Serviços Seguro e Flexível para Ambientes Ubíquos

Rodolfo S. Antunes[1], **Marinho P. Barcellos**[2,3] **(orientador)**

[1]UNISINOS – Universidade do Vale do Rio dos Sinos
Av. Unisinos, 950 – São Leopoldo, RS

[2]PPGC – Programa de Pós-Graduação em Computação
UFRGS – Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500 – Campus do Vale – Bloco IV – Porto Alegre, RS

[3]PUCRS – Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681, Prédio 32 – Porto Alegre, RS

rodolfo.s.antunes@gmail.com, marinho@acm.org

Abstract. *Flexible Secure Service Discovery (FSSD) is a service discovery protocol for ubiquitous environments. Designed to offer flexibility to the user, it allows the configuration of the collaboration, security and privacy levels used in its operation. This work presents the implementation of a FSSD prototype, with a simulation framework, used to analize the behavior of the protocol. Results show that the protocol satisfy the requirements of its specification.*

Resumo. *Flexible Secure Service Discovery (FSSD) é um protocolo de descoberta de serviços para ambientes ubíquos. Projetado para oferecer flexibilidade ao usuário, ele permite a configuração dos níveis de colaboração, segurança e privacidade durante sua operação. Este trabalho apresenta a implementação de um protótipo do FSSD, através de um* framework *de simulação, utilizado para a análise do comportamento do protocolo. Os resultados mostram que o protocolo atende aos requisitos de sua especificação.*

1. Introdução

Um dos principais objetivos da computação ubíqua é oferecer serviços computacionais de maneira transparente para usuários e dispositivos. Um *protocolo de descoberta de serviços* é um componente importante neste contexto, pois abstrai do usuário questões de configuração durante interações com outros dispositivos. O processo de descoberta pode ser abstraído no casamento entre consultas por um serviço e anúncios do mesmo serviço.

O *Flexible Secure Service Discovery* (FSSD) é um protocolo de descoberta de serviços cujo projeto é centrado em dois aspectos principais: segurança e flexibilidade. Os mecanismos do protocolo operam de maneira distribuída, de forma que seu uso se torna viável em um ambiente ubíquo, onde não há uma infraestrutura estática de rede. A descoberta é realizada de acordo com as escolhas do usuário dentro do *trade-off* entre colaboração, segurança e privacidade. Este *trade-off* pode ser ajustado dinamicamente, conforme o contexto onde a descoberta de serviços é utilizada.

O objetivo deste artigo é apresentar os principais resultados obtidos com o trabalho realizado como bolsista de Iniciação Científica do CNPq[1] entre 2007 e 2008. Como parte

[1]Rodolfo Stoffel Antunes: aluno com apoio do CNPq

deste trabalho, acompanhei as discussões que levaram à concepção do protocolo FSSD [Moschetta et al. 2008], participei do projeto da avaliação daquele protocolo, implementei a simulação e realizei experimentos com a mesma. Além disso, contribui na análise dos resultados. Os experimentos viabilizaram a análise do protocolo em um ambiente controlado, verificando se os princípios que guiaram seu projeto seriam satisfeitos.

A Seção 2 destaca os aspectos que diferenciam o FSSD de outros trabalhos na área. A Seção 3 apresenta, de forma resumida, os detalhes teóricos do protocolo projetado. A Seção 4 apresenta a simulação desenvolvida para observar o comportamento do protocolo, e os resultados através dela obtidos são apresentados na Seção 5. Finalmente, na Seção 6, são feitas algumas considerações sobre o trabalho realizado.

2. Trabalhos Relacionados

A área da descoberta de serviços é ampla, apresentando trabalhos em redes de diversos tipos [Zhu et al. 2005a, Edwards 2006, Mian et al. 2006]. Existem trabalhos desenvolvidos para ambientes com redes estruturadas, onde há um forte controle administrativo, como por exemplo o Ninja SDS [Czerwinski et al. 1999]. Um dos requisitos básicos de um ambiente ubíquo, porém, é uma arquitetura descentralizada, independente de infraestrutura [Satyanarayanan 2001].

Nesse sentido, diversos trabalhos propõem mecanismos distribuídos para o tratamento da segurança. Por exemplo, [Almenarez et al. 2006] propõe um mecanismo baseado em redes de confiança e certificados para a autenticação de pares. Em [Wishart et al. 2005] é apresentado um mecanismo para a manutenção de um índice da reputação dos serviços presentes na rede. Em relação à proteção da privacidade dos pares, [Zhu et al. 2005b] utiliza filtros Bloom. Em [Trabelsi et al. 2006] as pesquisas são cifradas utilizando os atributos do serviço como chave pública, de forma que a consulta será exposta apenas aos provedores que possuem o serviço procurado. Os protocolos citados, porém, têm a característica de favorecer apenas um requisito principal em seu projeto: ou segurança, ou privacidade.

O FSSD utiliza uma rede de *overlay* em nível de aplicação, baseada em confiança, para garantir a segurança durante a propagação de consultas e anúncios. A informação de confiança presente na rede é utilizada durante a descoberta, e posteriormente durante a seleção dos serviços encontrados. Também é implementado um mecanismo de controle de exposição, responsável por garantir a privacidade das informações que são encaminhadas a cada par. Esse mecanismo suprime informações sensíveis de acordo com a confiança existente entre os pares em comunicação.

Há uma relação direta entre o grau de colaboração dos pares e a quantidade de serviços encontrada por uma consulta. Ou seja, quanto mais serviços consultados e anunciados, maior será a quantidade de casamentos que irão ocorrer. Do mesmo modo, quanto mais informação estiver disponível nas mensagens, maior será a chance de casamento.

Por outro lado, uma colaboração maior implica uma exposição maior de informações aos outros pares da rede, pondo em risco a privacidade do usuário. O risco à segurança dos usuários, neste caso, é ampliado no contexto da computação ubíqua, onde a possível falta de um domínio administrativo dificulta o controle de ataques provenientes de pares maliciosos. O uso de redes de confiança permite a mitigação dos efeitos negativos na segurança e privacidade, devido à colaboração. A privacidade do usuário pode ser mantida através da troca de informações sensíveis apenas com pares confiáveis, com os

quais é compartilhada uma chave privada. O uso de redes de confiança, porém, pode levar à perda de privacidade, caso a identidade de um par seja associada às recomendações necessárias ao gerenciamento de confiança [Singh and Liu 2003].

Sendo assim, é possível estabelecer um *trade-off* entre os requisitos de colaboração, segurança e privacidade, que acabam se tornando objetivos conflitantes a serem buscados no projeto de um sistema de descoberta de serviços. Os trabalhos anteriormente citados favorecem um desses aspectos no projeto do protocolo. No projeto do FSSD, a flexibilidade foi escolhida como principal requisito. O FSSD permite que que os pares determinem o grau de colaboração, segurança e privacidade em tempo de execução. Os *trade-offs* podem ser definidos para cada serviço e consulta, já que os requisitos do usuário podem variar conforme o contexto em que o serviço de descoberta é utilizado.

3. *Flexible Secure Service Discovery*

O FSSD é um protocolo de descoberta de serviços em nível de aplicação, que opera sobre uma rede *overlay* construída a partir de uma rede de confiança. A topologia desse *overlay* é independente da rede física subjacente.

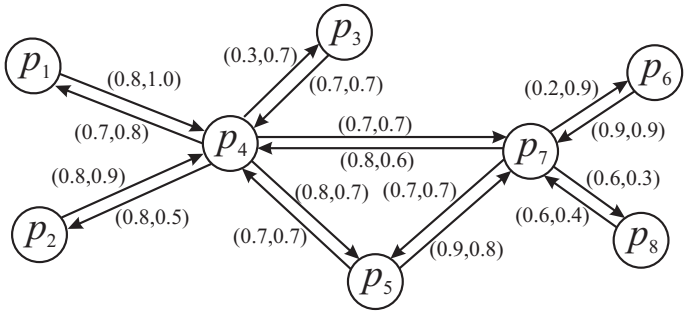

Figura 1. Exemplo de uma rede de confiança

A Figura 1 apresenta um exemplo de rede de confiança utilizada pelo FSSD. Cada vértice representa um par, enquanto as arestas representam as relações de confiança existentes entre os pares. O grau de confiança de um par p_i em um outro par p_j é denotada por $R(i,j) = (t_{ij}, c_{ij})$, onde $t_{ij} \in [0,1]$ corresponde à confiança de p_i em p_j, e $c_{ij} \in [0,1]$ é o grau de certeza sobre t_{ij}. A existência de uma relação $R(i,j)$ implica a existência da relação inversa, $R(j,i)$. Relações de confiança são construídas apenas com base em interações diretas entre os pares da relação, e permitem que os pares estabeleçam um canal seguro de comunicação entre eles.

Os pares utilizam um controle de exposição para restringir a disseminação de informações sensíveis. Consultas e anúncios são classificados como *informações de serviço*, formadas por um cojunto de informações a serem expostas, e denotadas por I. I^s é um subconjunto das informações contidas em I, onde s representa o nível de sensibilidade das informações contidas em I^s, com $s \in \{1,2,3\}$. Quanto maior for s, maior é a sensibilidade das informações contidas no subconjunto. Cada par define o nível mínimo de confiança necessário para a exposição das informações contidas em I^s aos outros pares. Este parâmetro é denotado por $T^s_{req} = (t_{req}, c_{req})$ e reflete a *prudência* de um par na rede. Quanto maior o valor de T^s_{req}, maior será a prudência do par, implicando uma maior privacidade, e uma menor colaboração.

Se um par p_i deseja enviar uma mensagem a um par p_j, é necessário que exista um canal seguro entre p_i e p_j, e $R(i,j) \geq T^s_{req}$, indicando que p_i tem confiança suficiente em

p_j para expor I^s. Do mesmo modo, se p_j encaminhar I^s para um outro par p_k, deve haver um canal seguro entre p_j e p_k, e $O(i,k) \geq T^s_{req}$. $O(i,k)$ é a opinião transitiva formada pela combinação de $R(i,j)$ com $O(j,k)$, através do uso de operadores de concatenação e consenso, como descrito em [Theodorakopoulos and Baras 2006]. Caso p_k não satisfaça o requisito T^s_{req}, p_j testará T^{s-1}_{req}, o que representa o envio de uma mensagem com informações suprimidas. Caso p_k atenda ao requisito T^{s-1}_{req}, p_j envia para p_k um subconjunto formado pela união das informações de menor sensibilidade, ou seja, I^{s-1}. Deste modo, as informações sensíveis de p_i, contidas no subconjunto I^s da mensagem, são protegidas.

O FSSD faz uso de quatro tipos básicos de mensagens: anúncios, consultas, recomendações e casamentos, conforme descrito a seguir. Anúncios e consultas são as principais mensagens utilizadas, possuindo os mesmos campos. O campo *message trace* contém o caminho percorrido pela mensagem na rede. Essa informação é utilizada para evitar a ocorrência de ciclos de roteamento, e para permitir que respostas sejam disseminadas através do caminho reverso. O campo *trust transitive chain information* contém a opinião transitiva acumulada pelos pares que encaminharam a mensagem. Essa informação é utilizada pelo controle de exposição em conjunto com o campo *exposure control requirements*, que contém os valores de T^s_{req} necessários para que a mensagem seja propagada. Finalmente, há o campo *service information record*, ilustrado pela Figura 2.

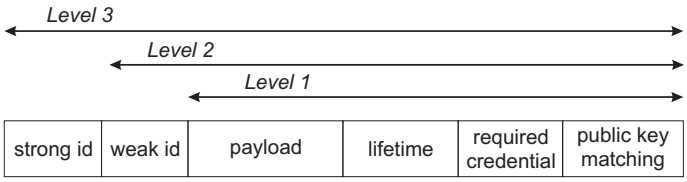

Figura 2. *Service information record* e níveis de sensibilidade.

O campo *service information record* é composto por um conjunto de informações, cada uma delas associada a um determinado nível de sensibilidade $s \in \{1, 2, 3\}$.

Nível 3 Contém o *strong id* de um par, que é a identidade formada por um endereço de rede (tal como um endereço MAC), utilizada para estabelecer um canal seguro de comunicação entre dois pares. Essa informação permite a identificação direta de um par na rede, e portanto requer uma exposição prudente.

Nível 2 Contém o *weak id* de um par, que é a identidade utilizada pelo gerenciamento de confiança. Essa informação não permite a identificação direta de um par na rede, e pode ser mais amplamente exposta.

Nível 1 Contém as informações que não permitem a identificação do par. O campo *payload* contém o tipo e os atributos do serviço anunciado ou consultado pela mensagem. O campo *lifetime* contém o tempo que a mensagem permanece válida quando armazenada por um par. O campo *required credential* descreve as credenciais necessárias para a ocorrência de um casamento. O campo *public key matching* possui a chave pública para a cifragem de mensagens de resposta.

A Figura 2 também ilustra os possíveis subconjuntos de informações que podem ser formados através da supressão dos dados, que poderá ocorrer durante o encaminhamento das mensagens através da rede. Caso a verificação das credenciais falhe para todos os níveis de sensibilidade, a mensagem não será encaminhada para o próximo par.

O terceiro tipo de mensagem presente no FSSD é a recomendação. Esta mensagem possui a mesma estrutura de uma mensagem de anúncio ou consulta, mas ao invés de

um *service information record*, possui um campo onde são listados os valores das relações de confiança $R(i, j)$ de um par. Esses valores são enviados aos outros pares para serem utilizados como recomendações. Essa informação é utilizada para aumentar a precisão dos valores das opiniões $O(i, j)$, calculadas durante o encaminhamento de mensagens na rede. Por se tratar de uma informação sensível, a disseminação de recomendações está sujeita aos mesmos critérios de prudência empregados para as consultas e anúncios.

O quarto tipo de mensagem, o casamento, é enviado quando ocorre o casamento de uma consulta com um anúncio de serviço. Esta mensagem possui, nos dois primeiros campos, os *service information record* contidos nas mensagens de anúncio e consulta que resultaram no casamento, além de um campo indicando se o casamento ocorreu no provedor do serviço consultado. O casamento das mensagens poderá ocorrer tanto no provedor do serviço, quanto em um par em que ambos provedor e cliente confiem. Este último caso é denominado casamento *in-network*. Quando um casamento *in-network* ocorre, a mensagem de casamento resultante conterá um campo com a opinião $O(i, p)$ que o par intermediário, responsável pelo casamento, possui sobre o par provedor do serviço.

4. Modelo de Simulação

Uma das principais características de um ambiente ubíquo é a sua dinamicidade. A avaliação do protocolo em um ambiente real, através de sua implementação em um conjunto de dispositivos móveis, permite uma análise completa das características do protocolo. Esse método, porém, introduz um conjunto elevado de variáveis que devem ser consideradas durante os experimentos, dada a dinamicidade do ambiente para qual o protocolo foi projetado. Tal fato leva ao mascaramento das propriedades básicas do protocolo, que devem ser avaliadas em uma fase inicial de implementação.

Por esse motivo, optou-se por utilizar simulações para avaliar as características básicas do protocolo, a fim de verificar se os requisitos levantados são atendidos pelo projeto proposto. A simulação foi desenvolvida através de uma extensão ao Simmcast [Muhammad and Barcellos 2002], um *framework* para a simulação de protocolos.

No Simmcast, os pares presentes em uma rede são representados através de um nodo. Cada nodo, por sua vez, possui um conjunto de serviços em execução denominados *threads*. Essas duas entidades são respectivamente mapeadas para as classes `Node` e `NodeThread`. A implementação da simulação, então, requer a extensão dessas duas classes para que o comportamento do protocolo a ser simulado seja incorporado. No caso do FSSD, cada nodo representa um dispositivo que irá tanto anunciar quanto buscar serviços presentes em outros nodos. Cada nodo terá a lógica do FSSD implementada em uma *thread*, representando a descoberta como um dos serviços em execução no nodo.

Cada nodo pode ter uma ou mais ligações com outros nodos na rede, sendo cada ligação representada por um caminho. Caminhos representam canais de comunicação unidirecionais, que não oferecem garantias de entrega ou ordenamento. Comparando-se uma rede do Simmcast com um grafo, cada nodo pode ser mapeado para um vértice, enquanto cada caminho pode ser mapeado para uma aresta direcionada. Ao criar-se um caminho entre dois nodos, é possível especificar a largura de banda, a latência, e a taxa de perda presentes no canal. Nas simulações realizadas, define-se que todos os caminhos representam conexões TCP com uma largura de banda efetiva de 10 Kbit/s, uma latência de 10 ms e que não há perda de pacotes.

O Simmcast também implementa um tipo especial de nodo, utilizado para atuar

como concentrador de conexões, como um *hub* ligando os diversos nodos da rede. Na simulação do FSSD, utiliza-se um concentrador, ligado através de um canal bidirecional com cada um dos nodos da rede. Este canal é formado por um par de caminhos entre o nodo e o concentrador. Deste modo, todas as simulações têm, em nível físico, uma rede interligada através de uma topologia do tipo estrela, permitindo que um nodo possa se comunicar com qualquer outro nodo da rede.

A comunicação entre os pares, porém, é restrita pelo FSSD através da rede de confiança estabelecida entre os pares. Conforme mencionado na seção 3, a rede de confiança é um *overlay* em nível de aplicação, estabelecido e gerenciado pelo protocolo FSSD. Uma rede *ad hoc*, cenário característico em ambientes ubíquos, possui uma formação dinâmica, onde pode ser necessário rotear mensagens entre diversos nodos para que esta atinja o seu destino. Em uma rede como a descrita, é esperada a entrada e saída de pares na rede com certa frequência. Entretanto, como um primeiro passo parte de uma estratégia para entendimento progressivo do comportamento do protocolo, foram considerados apenas casos estáticos. Essa rede é configurada antes da execução da simulação, e permanece inalterada até seu final. Deste modo, torna-se possível isolar a análise da influência da topologia da rede de confiança nos mecanismos do FSSD.

Redes de confiança apresentam características similares às de uma rede *small world*, tais como alta clusterização, baixa distância média entre dois vértices, e uma escala de comprimento logarítmica [Capkun et al. 2002]. Sendo assim, o algoritmo proposto em [Capkun et al. 2002] é utilizado para gerar um grafo com as propriedades descritas, utilizado como base para estabelecer a rede de confiança utilizada nas simulações. A geração da topologia da rede de confiança é realizada através de um programa auxiliar, desenvolvido em conjunto com a simulação. A Figura 3 ilustra a representação gráfica de uma topologia criada através do gerador.

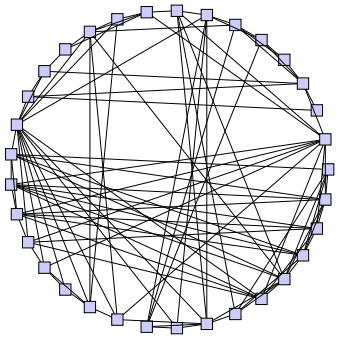

Figura 3. Exemplo de uma rede de confiança produzida pelo gerador de topologias

Uma aresta entre os pares p_i e p_j, neste caso, indica a existência das relações de confiança $R(i, j)$ e $R(j, i)$. Durante a criação da topologia, é atribuída a tupla de valores de confiança e certeza (t, c) a cada uma das relações existentes. Ambas variáveis podem assumir três valores: 0.1, que representa um baixo nível de confiança ou certeza; 0.5, no caso de um nível médio; e 0.9, indicativo de um alto nível.

Para simular o comportamento de diferentes tipos de usuários, em relação à opção entre colaboração ou segurança, exitem três *classes de segurança*. Essas classes definem o comportamento do par em relação à prudência na exposição de informações. Os pares da classe C_1 têm um alto requisito de prudência, e terão $T_{req} = (0.9, 0.9)$. Os pares da classe C_2 representam um grupo com requisitos médios de prudência, e terão $T_{req} = (0.5, 0.9)$.

Finalmente, os pares da classe C_3 representam uma classe pouco prudente, mais interessada em colaborar com a descoberta, tendo $T_{req} = (0.5, 0.5)$. Os pares presentes na simulação são uniformemente distribuídos entre essas três classes de segurança.

Em todos os experimentos há um conjunto de 10 tipos de serviços disponíveis na rede, uniformemente distribuídos entre os pares. Cada par é provedor de um serviço. Considerando que o número de pares é maior que 10, há múltiplas instâncias de um mesmo tipo de serviço presentes na rede, periodicamente anunciadas pelo par provedor. Ocasionalmente, um par escolhe um tipo de serviço, e então envia uma mensagem de consulta para este tipo, aguardando um tempo definido. Nesse tempo, o par coleta as mensagens de casamento recebidas. As mensagens de casamento recebidas para o serviço pesquisado são contabilizadas, sendo utilizadas no cálculo da eficácia da descoberta.

5. Resultados

A análise do protocolo foi direcionada no sentido de encontrar respostas para duas questões fundamentais, formuladas a partir dos requisitos definidos para o protocolo. Estas questões resultaram em dois conjuntos distintos de experimentos. Para cada conjunto, foram executadas 20 repetições dos experimentos, para gerar resultados estatisticamente válidos. Em cada repetição, uma topologia distinta é utilizada para a rede de confiança.

Q1: Em relação ao *trade-off* entre colaboração, segurança e privacidade, qual é o impacto dos requisitos de confiança T_{req} na eficácia do protocolo e na segurança e privacidade dos pares?

Q2: Qual o impacto provocado no processo de descoberta por alterações na topologia da rede de confiança?

5.1. Grupo 1: Parâmetros de segurança

Neste grupo, foram variados os parâmetros de segurança impostos pelos pares. Como as classes de segurança definidas possuem um valor fixo de confiança, no primeiro grupo de experimentos T_{req} é variado de acordo com um valor denotado por Δ_t. O objetivo desse método é simular a mudança de comportamento do usuário em relação a prudência adotada nas pesquisas, além de permitir a avaliação das alterações de comportamento nas três classes de segurança de maneira conjunta. A variação de Δ_t foi limitada em $[-0.4; 0.4]$, pois estes valores permitiram a transição dos parâmetros de prudência de um nível pouco restritivo (no caso de $\Delta_t = -0.4$), até um nível de alta prudência (no caso de $\Delta_t = 0.4$). O resultado final da variação é sempre mantido dentro do intervalo $[0.0; 1.0]$, para garantir a consistência dos parâmetros de prudência.

A principal métrica observada é a eficácia da descoberta, avaliada pela razão entre o número de mensagens de consulta geradas pelos pares, e o número de mensagens de casamento obtidas em respota às consultas geradas. Nessa métrica, existe a influência da prudência definida para as mensagens de consulta, que será variada por Δ_t, e também da prudência das mensagens de anúncio, fixa de acordo com a classe de segurança.

A Figura 4 apresenta os resultados para a métrica de eficácia. O eixo horizontal do gráfico mostra o nível de variação dos valores originais das classes de segurança (Δ_t). O eixo vertical apresenta a eficácia da descoberta. Cada curva representa as pesquisas realizadas para os anúncios produzidos por pares de uma das classe de segurança.

Observando-se inicialmente o cenário para os valores originais de T_{req} ($\Delta_t = 0$), é possível notar que os anúncios da classe C_3 geraram os maiores níveis de casamento,

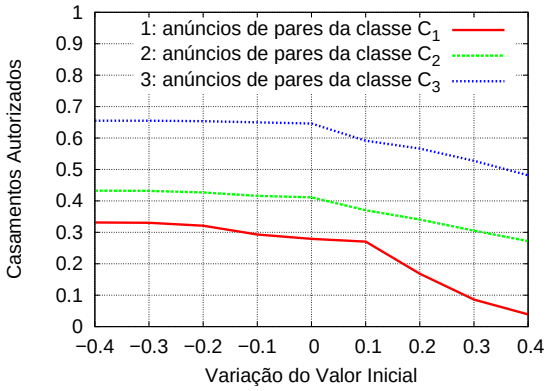

Figura 4. Eficácia da descoberta variando T_{req} das consultas

ficando próximo aos 65%. Tal resultado era esperado, pelo fato de C_3 ser a classe de menor prudência na exposição de suas informações. A classe de maior prudência, C_1, também como esperado, teve o menor número de casamentos obtidos, ficando próxima dos 30%. Os anúncios da classe C_2, por sua vez, resultaram em uma eficácia de pouco mais de 40% neste caso.

Em geral, as curvas apresentam um comportamento constante para os casos onde $\Delta_t \leq 0$. A partir deste ponto, porém, ocorre um declínio da eficácia em todos os casos. Os anúncios da classe C_1, no caso $\Delta_t = 0.4$, apresentam uma eficácia menor que 10%, uma queda de mais de 20% se comparado com o caso inicial ($\Delta_t = 0$). As outras curvas, neste mesmo caso, tiveram um declínio menor, próximo de 15%.

Finalmente, é possível observar que a diferença média da eficácia entre as classes de maior e menor prudência varia em torno de 30% a 40%, mantendo-se constante ao longo das curvas. Neste caso, cabe ao usuário definir se sua prioridade, em relação ao anúncio de seus serviços, é manter sua privacidade dentro da rede, ou oferecer o serviço ao maior número de pares possível. Este resultado demonstra que o protocolo é flexível, atendendo ao seu principal requisito de projeto.

5.2. Grupo 2: Rede de confiança

O segundo grupo de experimentos demonstra a influência do número de pares no processo de descoberta. Neste caso, os valores das classes de confiança foram mantidos inalterados, e o número de pares presentes na rede durante a simulação foi variado.

A Figura 5 apresenta, no eixo horizontal, o número de pares presentes na rede durante o experimento. No eixo vertical, novamente, é apresentada a eficácia da descoberta, com cada curva representando os anúncios de pares de uma classe de segurança.

Novamente, é possível observar uma diferença de aproximadamente 40% entre a eficácia apresentada pelos anúncios das classes de maior e menor prudência, que se mantém constante. As curvas indicam, também, que há uma tendência de queda da eficácia conforme o número de pares na rede é ampliado. Isso pode ser explicado pelo fato de que um maior número de pares aumenta a distância entre provedores e clientes, fazendo com que um número menor de casamentos ocorra. No pior caso observado, dobrando-se o número de pares de 512 para 1024, a queda na eficácia foi de aproximadamente 5%.

Quando existem poucos pares na rede, há uma influência menor da distância entre clientes e provedores nos resultados. Neste caso, a existência de mais relações de confi-

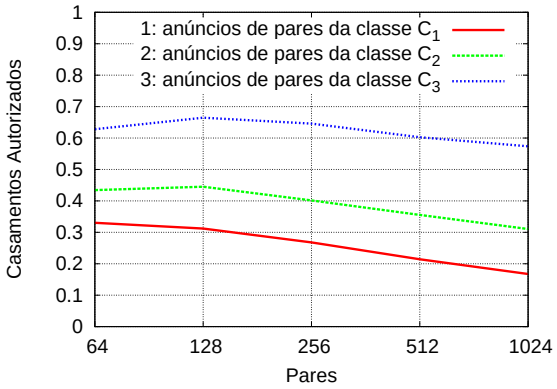

Figura 5. Eficácia da descoberta variando o número de pares

ança aumentará a eficácia, pois haverá mais caminhos para a disseminação das mensagens pela rede. Tal fato pode ser observado no caso de 128 pares, para as classes C_2 e C_3, onde há um leve crescimento da eficácia, ao contrário do restante dos casos apresentados.

A partir dos resultados observados, conclui-se que o número de pares terá influencia na eficácia do protocolo de descoberta. Isso ocorre pois o casamento de anúncios e consultas depende das relações de confiança estabelecidas, e dos caminhos que devem ser percorridos pelas mensagens, que se modificam conforme o número de pares na rede é ampliado. A queda de 5% na eficácia do pior caso observado, constante ao longo dos experimentos, demonstra que o protocolo é escalável.

6. Considerações Finais

Este artigo apresentou a avaliação do protocolo de descoberta de serviços para ambientes ubíquos FSSD. Ele foi re-escrito e traz melhorias na apresentação do modelo de simulação e dos resultados obtidos, em relação ao trabalho em [Moschetta et al. 2008]. Um dos princípios do protocolo é a segurança das informações expostas à rede, obtida através de uma rede de confiança e um controle de exposição. Outro princípio é a flexibilidade, através definição de uma prioridade entre colaboração, privacidade e segurança.

A necessidade da avaliação da proposta do FSSD levou à implementação de um protótipo em forma de simulação, trabalho por mim desenvolvido durante meu trabalho como bolsista de Iniciação Científica do CNPq entre 2007-2008. A análise dos resultados mostrou que o protocolo é eficaz na descoberta de serviços, mesmo com o uso da supressão de informações, para garantir a privacidade do usuário.

Os resultados obtidos neste trabalho foram utilizados na definição de melhorias no projeto inicial do FSSD e serviram como base para preparação de um artigo para periódico. Como trabalhos futuros, a simulação será estendida para suportar uma rede de confiança dinâmica, além de outros parâmetros tratados de maneira estática neste trabalho. Também pretende-se implementar o protocolo em um ambiente real, para avaliar suas propriedades quando utilizado em dispositivos móveis.

Referências

Almenarez, F., Marin, A., Diaz, D., and Sanchez, J. (2006). Developing a model for trust management in pervasive devices. In *Pervasive Computing and Communications*

Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on, pages 5 pp.+.

Capkun, S., Buttyan, L., and Hubaux, J. P. (2002). Small worlds in security systems: an analysis of the pgp certificate graph. In *Proceedings of the ACM New Security Paradigms Workshop*.

Czerwinski, S. E., Zhao, B. Y., Hodes, T. D., Joseph, A. D., and Katz, R. H. (1999). An architecture for a secure service discovery service. In *Mobile Computing and Networking*, pages 24–35.

Edwards, W. K. (2006). Discovery systems in ubiquitous computing. *Pervasive Computing, IEEE*, 5(2):70–77.

Mian, A. N., Beraldi, R., and Baldoni, R. (2006). Survey of service discovery protocols in mobile ad hoc networks. Technical report, Dipartimento di Informatica e Sistemistica "Antonio Ruberti", Università degli Studi di Roma "La Sapienza", Rome, Italy.

Moschetta, E., Barcellos, M. P., and Antunes, R. S. (2008). Flexibilizando graus de colaboração, segurança e privacidade na descoberta de serviços em ambientes ubíquos. *XXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2008)*, pages 1–14.

Muhammad, H. H. and Barcellos, M. P. (2002). Simulation group communication protocols through an object-oriented framework. In Sc, S., editor, *35th Annual Simulation Symposium, ANSS 2001*, volume 1, San Diego, USA. SCS.

Satyanarayanan, M. (2001). Pervasive computing: vision and challenges. *Personal Communications, IEEE*, 8(4):10–17.

Singh, A. and Liu, L. (2003). Trustme: anonymous management of trust relationships in decentralized p2p systems. In *Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on*, pages 142–149.

Theodorakopoulos, G. and Baras, J. S. (2006). On trust models and trust evaluation metrics for ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 24(2):318–328.

Trabelsi, S., Pazzaglia, J. C., and Roudier, Y. (2006). Secure web service discovery: Overcoming challenges of ubiquitous computing. In *ECOWS '06: Proceedings of the European Conference on Web Services*, pages 35–43, Washington, DC, USA. IEEE Computer Society.

Wishart, R., Robinson, R., Indulska, J., and Josang, A. (2005). Superstringrep: reputation-enhanced service discovery. In *CRPIT '38: Proceedings of the Twenty-eighth Australasian conference on Computer Science*, pages 49–57, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

Zhu, F., Mutka, M. W., and Ni, L. M. (2005a). Service discovery in pervasive computing environments. *Pervasive Computing, IEEE*, 4(4):81–90.

Zhu, F., Zhu, W., Mutka, M. W., and Ni, L. (2005b). Expose or not? a progressive exposure approach for service discovery in pervasive computing environments. In *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pages 225–234, Washington, DC, USA. IEEE Computer Society.