

7 Trabalhos Relacionados

7.1. Utilização de SIP para Gerenciamento de Mobilidade

Todos os trabalhos que tratam do Gerenciamento de Mobilidade na camada de aplicação se referem diretamente ao protocolo SIP, não sendo apresentado, no melhor do nosso conhecimento, qualquer outro protocolo alternativo a este na literatura. Desde a publicação dos trabalhos que propuseram a solução original (Wedlund e Schulzrinne, 1999; Schulzrinne e Wedlund, 2000), diversos outros trabalhos exploraram aspectos como análise do tempo de handoff (i.e. tempo de migração entre sub-redes) e otimização desta solução para diferentes tipos de cenários (Nakajima, Dutta et al., 2003; Banerjee, Wu et al., 2004; Iyov e Noel, 2004; Kim, Kim et al., 2004; Banerjee, Das et al., 2005; Banerjee, Acharya et al., 2006).

Entretanto, apenas um único trabalho foi encontrado que apresentasse o desenvolvimento desta solução de Gerenciamento de Mobilidade. O trabalho de (Yeh, Wu et al., 2006) apresenta o projeto e desenvolvimento de uma arquitetura para Gerenciamento de Mobilidade na camada de aplicação com SIP muito semelhante à apresentada nesta dissertação. Ela contém uma API que é um User Agent SIP chamado de NCTU SIP UA (National Chiao Tung University SIP User Agent). Além da descrição do projeto e componentes da arquitetura de software do NCTU SIP UA, é também apresentado um estudo do atraso na realização de migrações entre diferentes sub-redes utilizando o IPv4 e IPv6. Uma outra diferença deste trabalho com relação a esta dissertação é o foco na transmissão de fluxos de voz utilizando o protocolo RTP, enquanto que nesta dissertação o foco é na utilização da solução implementada para suportar a mobilidade e desconexão de sistemas publish/subscribe, juntamente com um protocolo de transmissão confiável de dados via UDP. Além disso, as tecnologias e o projeto da arquitetura também apresentam diferenças, incluindo a linguagem de programação. A API NCTU SIP UA foi desenvolvida com na linguagem C de programação e com

APIs auxiliares, e.g. a API eXosip (The eXtended osip library) que possui funcionalidade equivalente da JAIN SIP, também desenvolvidas nesta linguagem. Por outro lado, nesta dissertação, a API SIP User Agent foi implementada com a linguagem Java de programação e utiliza APIs auxiliares (e.g. JAIN SIP) que também foram desenvolvidas nesta linguagem. A utilização da linguagem Java permite uma maior portabilidade, inclusive para dispositivos móveis portáteis.

7.2. Mobilidade em Sistemas Publish/Subscribe

Como já foi mencionado, este trabalho tem também como objetivo aplicar a solução de Gerenciamento de Mobilidade com o protocolo SIP a um sistema publish/subscribe para torná-lo capaz de suportar a mobilidade e desconexão de dispositivos. Esta Seção relaciona este trabalho com os demais trabalhos da literatura que tratam de mobilidade em sistemas publish/subscribe.

O trabalho de (Huang e Garcia-Molina, 2004) apresenta os principais aspectos a serem considerados quanto à adaptação de sistemas publish/subscribe para o cenário de mobilidade, tanto para sistemas com arquiteturas centralizadas como distribuídas. Suas observações com relação aos sistemas publish/subscribe de arquiteturas centralizadas são consideradas no escopo central deste trabalho. Dentre as principais observações, podemos destacar as que sugerem o correto estabelecimento dos componentes do sistema em um cenário de mobilidade (e.g. a necessidade do servidor de eventos ser estabelecido em um dispositivo na rede fixa), assim como as que apresentam uma introdução sobre mecanismos de enfileiramento de notificações para subscritores desconectados, sumarizações de notificações nas filas para resolver problemas de congestão do sistema e mecanismos para tratar da desconexão de publicadores de eventos, com enfileiramento e descarte de eventos diretamente na fonte (quenching). Eles, contudo, não investigam uma implementação e experimentação concreta das características abordadas e também qual a forma de Gerenciamento de Mobilidade a ser utilizada para garantir a conectividade entre os componentes do sistema.

Diversos outros trabalhos (Caporuscio, Carzaniga et al., 2003), (Cugola, Di Nitto et al., 2001), (Fiege, Gartner et al., 2003), (Mühl, Ulbrich et al., 2004), (Zeidler e Fiege, 2003) tratam da adaptação de sistemas publish/subscribe (e.g.

adaptações dos os sistemas Elvin, JEDI, REBECA, Siena, dentre outros) para redes móveis. A maioria dos sistemas publish/subscribe com suporte a mobilidade apresentados nesses trabalhos possui uma arquitetura distribuída. Um sistema publish/subscribe com arquitetura distribuída possui a vantagem de ser escalável para um grande número de consumidores e publicadores de informações. Com a importância dada ao aspecto distribuído dos principais sistemas, pode-se observar nos trabalhos acima mencionados que, na maioria dos casos, o foco é a mobilidade de dispositivos entre diferentes servidores de eventos (i.e. um dispositivo se desconecta de um Broker e se conecta a outro). Ou seja, os trabalhos propõem algoritmos distribuídos que os Brokers executam para realizar a sincronização de subscrições, fazer o roteamento e entrega correta de notificações, dentre outras operações a serem executadas quando um evento de mobilidade como este ocorre.

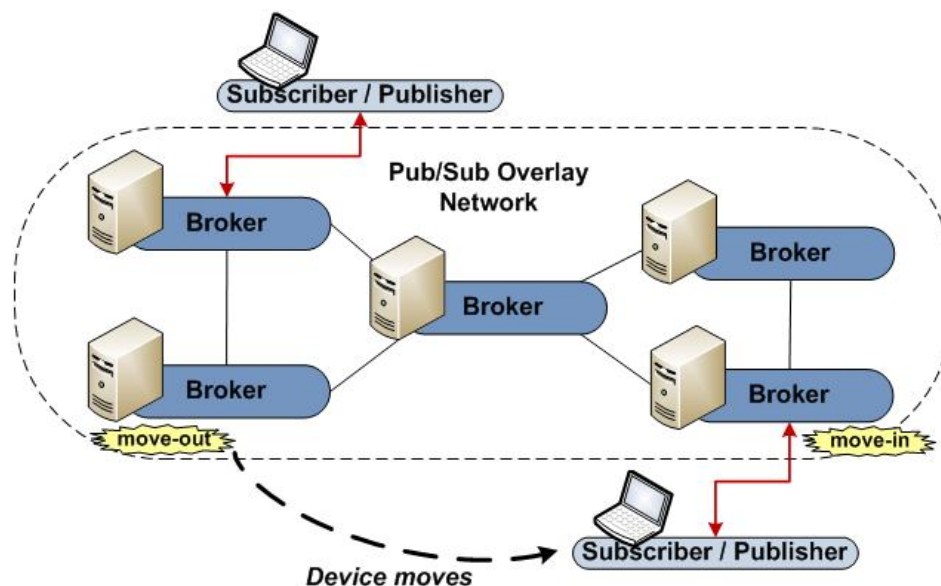


Figura 41 - Mobilidade em sistemas publish/subscribe com arquitetura distribuída.

Para suportar estas operações a maioria dos trabalhos propõe a criação de Proxys que fazem a intermediação da comunicação de clientes móveis com Brokers, oferecendo para os clientes operações de move-in (se anexar a um Broker) e move-out (para deixar um Broker). Quando um dispositivo detecta sua própria mobilidade ele chama a operação de move-out de seu Broker corrente e em seguida chama a operação de move-in do Broker para o qual se moveu. Estas operações permitem que os Proxys executem os algoritmos necessários para a migração com os devidos ajustes no sistema como um todo. No momento em que o cliente faz uma operação de move-out, ele copia suas subscrições para o Proxy

para que este receba as notificações em seu nome e as armazene para serem entregues quando ele se reconectar.

O que difere os trabalhos mencionados da abordagem deste trabalho não é a utilização do conceito de uma entidade intermediadora (Proxy) entre os clientes e o servidor, mas sim de como o conceito de mobilidade é abordado. Neste trabalho, a mobilidade de um dispositivo é considerada como a mudança de seu endereço IP, enquanto o dispositivo se move entre diferentes redes permanecendo conectado ao mesmo servidor de eventos.

Embora os trabalhos acima mencionados tenham o foco em arquiteturas distribuídas, muitas características abordadas por eles são utilizadas por este trabalho, visto que são comuns a aquelas que se aplicam à adaptação de sistemas centralizados para o ambiente móvel. Por exemplo, embora o ECI não tenha sido adaptado através da criação de um Proxy para o tratamento da mobilidade dos dispositivos, este conceito poderia ter sido utilizado para adaptar o único servidor de eventos da arquitetura centralizada para lidar com a mobilidade, o que não foi realizado apenas por decisões de projeto motivadas pela já existência de outro Proxy existente na arquitetura do middleware MoCA.

Proxys em geral se conectam a um Broker de forma estática, ou seja, para o servidor de eventos os Proxys são como clientes estáticos normalmente conectados. Os Proxys devem, portanto, implementar mecanismos que atendam aos requisitos de desconexão dos dispositivos, tal como um servidor centralizado, por exemplo, os requisitos apresentados no Capítulo 5. Assim, estes trabalhos serviram de contribuição para o levantamento dos requisitos de um sistema centralizado com suporte a mobilidade e desconexão. O trabalho de (Sutton, Arkins et al., 2001), por exemplo, apresenta uma extensão para o sistema Elvin que oferece mecanismos tais como persistência de notificações, sessões de usuários, recepção de notificações em diferentes dispositivos, recebimento não destrutivo de notificações e tempo de vida de subscrições. Ele, entretanto, não menciona que tipo de Gerenciamento de Mobilidade é utilizado para manter a conectividade entre os clientes e os Proxys, caso os dispositivos mudem de endereço IP por causa da mobilidade. O trabalho de (Mühl, Ulbrich et al., 2004) apresenta uma extensão para o sistema REBECA que oferece subscrições duráveis, políticas de enfileiramento de notificações, acesso a notificações antigas, limitações na taxa de entrega e limitações por conteúdo. Para enviar as

notificações, um Proxy utiliza uma janela deslizando sobre uma fila temporária de notificações onde guarda as notificações até que o cliente confirme sua recepção. No caso de desconexão de um cliente, quando este retorna, o número de seqüência da última notificação recebida é enviado pelo cliente e o Proxy continua a transmissão a partir da última notificação.