



MUSIC.LOCAL: UMA APLICAÇÃO DAS TECNOLOGIAS ZEROCONF

Paulo Leonardo Souza BRIZOLARA¹, Allyson Amílcar Angelus Freire SOARES, João Moreno VILAS BOAS³

¹Discente do curso Técnico em Informática– IFRN. e-mail: souzabrizolara@gmail.com;

²Professor de Informática – IFRN. e-mail: allyson.soares@ifrn.edu.br;

³Professor de Eletrônica - IFRN. e-mail: joao.vilasboas@ifrn.edu.br

RESUMO: A quantidade e diversidade de dispositivos computacionais no mercado tem crescido muito nos últimos anos. *Tablets* e *smartphones* se consolidam, enquanto surgem novos aparelhos, como televisores e relógios “inteligentes”. Estas condições têm levado a um novo cenário que estimula o uso destes dispositivos em conjunto. Para possibilitar a integração plena entre eles, no entanto, existem ainda diversos desafios. Um dos primordiais está na própria conexão entre estes dispositivos. Mesmo cenários básicos, como a troca de dois arquivos, exigem, muitas vezes, conhecimento técnico ou hardware especializado. As tecnologias Zeroconf proveem uma solução em redes IP para este desafio. Neste trabalho estas tecnologias são discutidas e comparadas a algumas alternativas existentes. Além disto, apresentamos o Music.local, um sistema resultante da aplicação da Zeroconf a um problema real: o compartilhamento de músicas em um ambiente residencial. Por fim, são expostos desafios e possibilidades para o uso destas tecnologias.

Palavras-chave: zeroconf, DNS-SD, Sistemas multi-dispositivo

MUSIC.LOCAL: AN APPLICATION OF ZEROCONF TECHNOLOGIES

ABSTRACT: The number and variety of computer devices on the market has grown in the last years. *Tablets* and *smartphones* are consolidating, while new appliances arise, like smart TVs and smart watches. These conditions have led to a new situation which encourages the joint use of these devices. However, to enable the full integration among them, there still many challenges. One of the primary is the connection between these devices. Even basic cases, like file sharing, require, many times, expert knowledge or specific hardware. Zeroconf technologies provide an IP solution to this challenge. This paper discuss these technologies and compares them to existing alternatives. Besides that, we present the Music.local, a system resulting from Zeroconf application to a real problem: the sharing of musics in a residential environment. Lastly, challenges and possibilities to the use of these technologies are exposed.

KEYWORDS: zeroconf, DNS-SD, multi-device Systems



INTRODUÇÃO

Estamos vivenciando um grande crescimento no número de dispositivos computacionais, tanto em quantidade quanto em diversidade. O número de dispositivos conectados já superou a casa dos 7 bilhões. Por sua vez, novos tipos de dispositivos tem chegado ao mercado, como *smart watches*, *smart TVs* e até mesmo dispositivos integrados ao carro, além de outros já consolidados, como *tablets*, *smartphones* e *notebooks* [LEVIN, 2014].

O acesso a estes aparelhos traz novas oportunidades e desafios para usuários e desenvolvedores. Pesquisas têm mostrado que estes dispositivos vêm sendo utilizados em conjunto, seja em sequência ou de forma simultânea, para completar determinadas tarefas, obter informações ou apenas por lazer [GOOGLE, 2012].

A escolha do aparelho a ser utilizado muitas vezes é determinada por fatores como local, tempo, objetivos ou mesmo estado de espírito. Estas possibilidades podem gerar experiências de uso mais ricas e novas possibilidades de atender aos usuários. Porém podem surgir também frustrações, se não houver suporte para transições naturais entre dispositivos nas aplicações utilizadas. Para desenvolvedores, por sua vez, lidar com esta gama de artefatos computacionais pode trazer grandes desafios e oportunidades de criar soluções inovadoras.

Um dos desafios primordiais para o desenvolvimento de sistemas “multi-dispositivo” é a interconexão entre estes equipamentos. Embora tecnologias como o Bluetooth facilitem a comunicação entre aparelhos, elas não se adequam tão bem às tecnologias de rede predominantes, baseadas em IP. Por isto costumam exigir hardwares e até soluções de software exclusivas. Uma alternativa para isto é a Zeroconf.

Zeroconf (*Zero Configuration Network*) consiste em um conjunto de tecnologias que buscam reduzir ou eliminar os entraves para a comunicação entre computadores e outros dispositivos de rede. Isto possibilita aos usuários destas tecnologias compartilhar conteúdo ou combinar capacidades de dispositivos, sem necessitar de conhecimentos técnicos ou do uso de serviços externos. A Zeroconf surgiu a partir de um grupo de trabalho do IETF (*Internet Engineering Task Force*), que buscou trazer capacidades presentes nas antigas redes AppleTalk, para as redes IP.

Estas tecnologias tem sido aplicadas em trabalhos presentes na literatura e no mercado, como forma de simplificar ou viabilizar a interconexão de dispositivos e aplicações. O DisplayCast, por exemplo, é um sistema para compartilhamento de telas entre múltiplos dispositivos. Ele utiliza a Zeroconf para a descoberta e conexão das diferentes aplicações do sistema, i.e. *Streamer*, *Player* e *Archiver* [CHANDRA e ROWE, 2012].

Fong et al. (2007), desenvolveram uma solução de *streaming* de conteúdo multimídia, a partir de aplicações de *streaming* pré-existentes, e aplicam as tecnologias Zeroconf para simplificar o uso destas.

Já Pedrosa et. al. (2011), aplicaram o DNS-SD para televisões digitais. Porém, voltado para a conexão de dispositivos de interação multimodal com aplicações da TV Digital.

Cortez et al. (2012), por sua vez, utilizaram também o DNS-SD em uma API para a integração de aplicações de “segunda tela” com aparelhos de TV da plataforma “Yahoo! Connected TV”.



O presente trabalho tem como objetivo o estudo das tecnologias Zeroconf, aplicado ao desenvolvimento de sistemas multi-dispositivo. Neste trabalho apresentamos, como resultado deste estudo, uma breve descrição das tecnologias analisadas e soluções similares disponíveis; e um sistema que aplica as tecnologias Zeroconf para o compartilhamento de músicas em um ambiente residencial.

MATERIAL E MÉTODOS

Zeroconf foi construída com base em padrões abertos, os mesmos já utilizados amplamente na internet. Ela acrescenta a eles, tecnologias em três diferentes camadas de rede: alocação de endereços IP, obtenção de nomes e a descoberta e publicação de serviços.

Em relação a primeira destas camadas, Zeroconf adota uma técnica de auto-atribuição de IPs, denominada IPv4 *Link Local Addressing* (IPv4LL). Esta permite conectar máquinas em uma rede IPv4 (o IPv6 possui suporte nativo a uma tecnologia similar) na ausência de servidores DHCP e sem necessidade de interferência pelos usuários.

No que diz respeito a atribuição de nomes à máquinas ou serviços em redes IP, o DNS é o protocolo padrão. O Zeroconf apresenta uma variação deste protocolo, denominado *Multicast DNS* (mDNS). Este permite atribuir nomes, no contexto de uma rede local, sem a necessidade de um servidor DNS e de forma automatizada (a resolução de conflitos é feita pelas próprias máquinas).

Tendo por base o protocolo DNS, padrão ou multicast, a Zeroconf introduz um novo protocolo, o DNS-SD (*DNS Service Discovery*). Este possibilita a publicação e descoberta automática de serviços em uma rede IP.

Este conjunto de tecnologias proveem uma fundação que torna possível a comunicação entre máquinas e entre aplicações de forma transparente aos usuários. Isto é, sem necessidade de configuração ou conhecimento técnico específico.

Outros protocolos oferecem recursos similares aos presentes na Zeroconf, como a descoberta e publicação de serviços. Alguns deles são apresentados a seguir.

O Service Location Protocol (SLP) foi um dos primeiros protocolos para descoberta de serviços a ser bem conhecido. Dispositivos que utilizam o SLP podem assumir um ou mais entre três diferentes papéis: Agentes de Usuário (AU), Agentes de Serviço (AS) e Agentes de Diretório (AD). O primeiro deles é quem realiza a busca por serviços, que são oferecidos na rede pelos Agentes de Serviço. Já os Agentes de Diretório são entidades responsáveis por agregar informações de serviços e responder a requisições dos Agentes de Usuário [MESHKOVA, 2008].

O SLP pode operar com ou sem a presença de ADs. Se um AD for detectado, os outros agentes utilizarão ele como centralizador das mensagens. Caso contrário, as mensagens serão trocadas diretamente entre os outros agentes [GUTTMAN, 1999].

A base para identificação de serviços no SLP é feita através de URL's com formato geral: *service:service-name:protocolname://hostname*. Adicionalmente, estes serviços podem possuir também atributos e um tipo. Estes dados são anunciados junto com a URL por Agentes de Serviço e podem ser utilizados por Agentes de Usuário para realizar consultas [MESHKOVA, 2008].



UPnP ou Universal Plug and Play abrange um outro conjunto de tecnologias que permitem a descoberta de dispositivos e serviços, bem como a comunicação automática entre eles. Ele é baseado também em tecnologias IP [UPnP, 2008]. No UPnP a comunicação ocorre através de 6 passos: endereçamento, descoberta, descrição, controle, escuta por eventos e apresentação.

Na primeira etapa o dispositivo obtém um número IP, via DHCP ou através do IPv4 *Link Local Addressing*. Já a descoberta de serviços utiliza o protocolo SSDP (*Simple Service Discovery Protocol*), o qual utiliza mensagens em multicast para permitir a descoberta de serviços na rede. Após esta etapa, o dispositivo que deseja utilizar os recursos de outro obtém uma URL de um documento XML com a descrição destes recursos. O dispositivo pode passar, então a: controlar o outro, utilizando o protocolo SOAP (*Simple Object Access Protocol*); passar a escutar eventos deste, através do protocolo GENA (*General Event Notification Architecture*); ou exibir ao usuário uma interface provida por este dispositivo, através de uma página web, por exemplo [MESHKOVA, 2008].

O UPnP é controlado por um fórum, que produz e publica protocolos para o controle de dispositivos. Essa é uma das características que diferencia o UPnP do Zeroconf. Enquanto o Zeroconf provê apenas a base para que dispositivos possam se comunicar, utilizando outros protocolos, o UPnP define quase todas as camadas de comunicação. Apesar disso, nada impede que ambos protocolos sejam utilizados por uma mesma aplicação [CHESHIRE, 2006].

Cada uma destas pilhas de protocolos possui características distintas que podem estar mais adequados a diferentes contextos de uso.

A Zeroconf, em especial, apresenta um conjunto de características que podem ser vantajosas para aplicações multi-dispositivo. Dentre elas, podemos citar: o uso de tecnologias IP bem estabelecidas como base, a não imposição de restrições sobre os protocolos de nível de aplicação, a relativa leveza em termos do uso de recursos computacionais, e ser formada por protocolos abertos.

Estas características podem prover benefícios, como: maior interoperabilidade entre tecnologias; redução de custos; relativa facilidade para adoção e implementação; e possibilidade de uso em sistemas embarcados e plataformas com altas restrições de recursos.

RESULTADOS E DISCUSSÃO

Com a intenção de explorar e estudar as tecnologias Zeroconf, foi desenvolvida a Music.local, uma aplicação que permite compartilhar músicas de um dispositivo com vários outros, no contexto de uma rede local. Esta aplicação utiliza uma arquitetura cliente-servidor RESTful. O servidor é o responsável por armazenar a coleção de músicas, enquanto nos clientes ocorre a reprodução do conteúdo.

Esta arquitetura (Figura 1) possibilita que vários dispositivos compartilhem uma mesma base de dados, e reflete a diferença existente entre os tipos de dispositivos, e as suas capacidades. Dispositivos móveis, por exemplo, usualmente apresentam capacidades de multimídia embutidas, além de dar aos usuários a possibilidade de se locomover no ambiente enquanto escutam música. Por outro lado, estes possuem recursos como memória e poder de

processamento limitados. Isto faz com que eles tendam a não ser um bom ambiente para a execução de um servidor deste sistema, embora apresentem boas características para um cliente. A arquitetura adotada, no entanto, não impede que uma mesma máquina execute um programa cliente e um servidor simultaneamente.

Cada servidor oferta, através do protocolo HTTP, uma interface para que outras aplicações (os clientes) possam obter as músicas conhecidas por ele, bem como informações sobre elas. Estas informações são dispostas utilizando o formato JSON (*JavaScript Object Notation*). O JSON é um formato de texto utilizado principalmente para a troca de dados entre aplicações, e tem sido muito utilizado em serviços Web. Ele é baseado na notação utilizada pela linguagem Javascript para descrição de objetos.

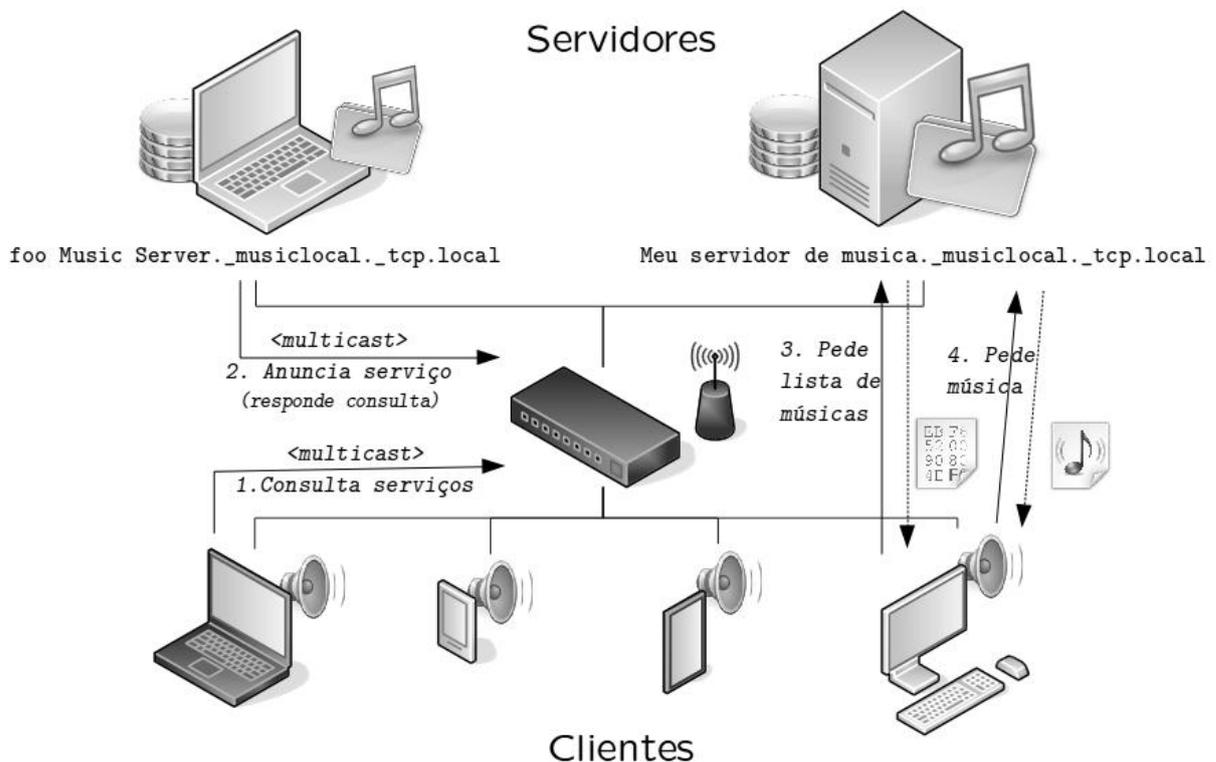


Figura 1. Arquitetura cliente/servidor da aplicação Music.local. (IFRN, 2015)

O conjunto de operações que o sistema oferta para seus clientes, aliado ao formato e à definição dos possíveis conteúdos, i.e. sua interface, definem um novo serviço. Assim, para anunciá-lo na rede, através do DNS-SD, um novo tipo de serviço foi utilizado: “musiclocal”.

No DNS-SD o nome do tipo de serviço é utilizado em conjunto com o nome de domínio, protocolo de transporte e um nome de instância (legível por usuários) para identificar um serviço. A combinação destes nomes resulta em um nome único com o seguinte formato: “<nome da instância>.<tipo do serviço>.<protocolo de transporte>.<domínio>”. Assim, uma instância do Music.local com nome “Meu servidor”,



X CONNEPI

X Congresso Norte Nordeste de Pesquisa e Inovação

registrado em um domínio local (utilizando o mDNS) teria o seguinte nome completo: “Meu servidor._musiclocal._tcp.local”.

Durante a inicialização, o servidor anuncia através do DNS-SD uma nova instância de serviço do tipo “musiclocal”. O nome da instância do serviço pode ser definido pelos usuários. Caso contrário, o sistema cria um nome padrão com base no nome da máquina (*hostname*). A porta utilizada também pode ser definida pelos usuários, mas por padrão o servidor obtém uma porta dinâmica. Como o serviço é anunciado via DNS-SD, isto não impõe dificuldades para se conectar ao servidor e, por sua vez, permite a execução de vários serviços em uma mesma máquina sem conflito (cada um utilizando uma porta diferente).

Para que os clientes possam reproduzir músicas, eles precisam primeiramente localizar e se conectar a um servidor. Para isto, eles utilizam o protocolo DNS-SD para buscar todos os servidores “musiclocal” na rede e exibi-los ao usuário. Até que este escolha um dos servidores disponíveis ou saia da aplicação, ela continua escutando na rede para determinar a entrada ou saída de servidores.

Após conectar-se, um cliente pode obter a lista de músicas disponíveis no servidor. Os arquivos de música, no entanto, são obtidos em demanda, para reduzir o tráfego de dados. Após serem carregados pela primeira vez, estes arquivos são mantidos em um cache no disco.

A lista de músicas no servidor é definida a partir de um “diretório de músicas”, que é determinado pelos usuários. O servidor realiza automaticamente a varredura de todos os arquivos de música, encontrados abaixo deste diretório (incluindo subpastas), adicionando-os à coleção. A figura 2 apresenta um diagrama de sequência da aplicação.

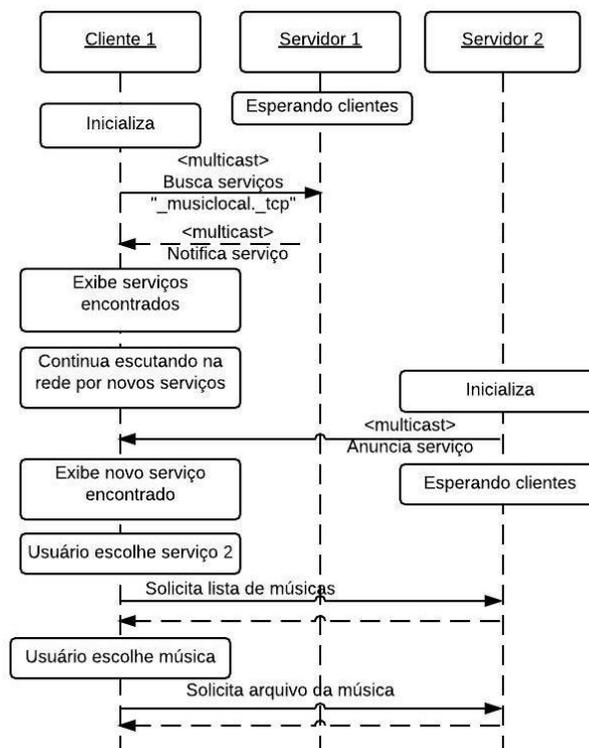


Figura 2. Diagrama de sequência do Music.local. (IFRN, 2015)



O servidor do Music.local foi implementado como uma aplicação Java, com uma interface de linha de comando. O suporte ao protocolo DNS-SD foi obtido utilizando a biblioteca JmDNS, uma implementação puramente em java do DNS-SD e mDNS. Esta aplicação pode ser executada nos principais sistemas operacionais *desktop*: Linux, Mac OS e Windows.

Atualmente, existe apenas uma implementação de cliente, na plataforma Android, suportando dispositivos com versão 2.3 (Gingerbread) ou superior. Embora, a plataforma Android ofereça uma implementação do protocolo DNS-SD, chamada de Network Service Discovery (NSD), esta suporta apenas dispositivos com versão igual ou superior a 4.1 (Jelly Bean). Por isto, optamos pelo uso da biblioteca JmDNS também.

Como apresentado anteriormente, Zeroconf provê uma base para o desenvolvimento de aplicações “multi-dispositivo”. Ela é uma solução leve, do ponto de vista computacional, mas também fácil de ser integrada à aplicações. Pois é baseada em tecnologias amplamente utilizadas, como o IP e o DNS, e não impõe restrições sobre o protocolo a nível de aplicação. Estas características proveem benefícios para desenvolvedores e usuários.

Apesar disto, existem fatores que dificultam ou limitam a aplicação destas tecnologias. Alguns destes estão relacionados à limitações na versão 4 do protocolo IP, mas que já apresentam soluções no IPv6. Tais como: o suporte à multicast, que é obrigatório no IPv6, mas não é adotado por todos os dispositivos de rede no IPv4; e o suporte a endereços IP de Link Local, que já é adotado por padrão no IPv6, mas exige um protocolo especial para o IPv4, o IPv4 *for Link Local Address*, discutido anteriormente.

Outros desafios estão relacionados à escalabilidade do Zeroconf. O DNS Unicast pode ser utilizado para publicar e localizar serviços globalmente. Entretanto, a forte hierarquia na estrutura do DNS e o custo para manter um servidor, exigem uma estrutura mais robusta ou custosa para manter aplicações que desejem utilizar o DNS-SD além do link-local. Isto pode dificultar o desenvolvimento de um conjunto de sistemas que busquem utilizar as tecnologias Zeroconf entre os escopos de link-local e intercontinental. Por exemplo, em ambientes corporativos ou universitários.

Possivelmente, um dos maiores desafios para o uso dos protocolos Zeroconf, ou similares, está no aprendizado e adoção deles em larga escala. Daí a necessidade de divulgar estas tecnologias, assim como realizar novos estudos e desenvolver aplicações com base nelas.

CONCLUSÕES

Este trabalho apresentou o resultado dos estudos feitos sobre a pilha de protocolos Zeroconf, bem como outras soluções alternativas, i.e. UPnP e SLP. Foram apresentadas as três tecnologias base da Zeroconf, com destaque para os protocolos adicionados por ela: IPv4 *for Link Local*, Multicast DNS e DNS Service Discovery.

Neste trabalho também foi introduzido o Music.local, uma aplicação para permitir o compartilhamento de músicas, de um host para um conjunto de clientes, em um ambiente residencial. Este sistema apresenta ainda um conjunto de limitações, porém ele demonstra como o Zeroconf pode ser utilizado para permitir o desenvolvimento de aplicações multi-



dispositivo. Trabalhos futuros poderiam expandir este sistema para, por exemplo: permitir o streaming de música em tempo real; criar sistemas de som através da reprodução sincronizada em múltiplas caixas de som; e a integração com tocadores de música *open-source* ou comerciais.

As tecnologias Zeroconf, por sua vez, apresentam vários campos em que podem ser aplicadas em trabalhos futuros. Tais como a automação residencial, a construção de sistemas multimídia com múltiplos dispositivos, bem como outros sistemas distribuídos. Além disto, estudos mais aprofundados podem ser conduzidos para, dentre outras coisas: analisar o uso de Zeroconf e soluções alternativas em aplicações do mercado; identificar limitações e oportunidades para estas tecnologias; e expandir os seus protocolos para facilitar seu uso além do escopo de *link-local*.

REFERÊNCIAS

- CHESHIRE, S. **Zero Configuration Networking (Zeroconf)**. 2006. Disponível em: <<http://www.zeroconf.org/>>. Acesso em: 06 mar. 2015.
- GOOGLE. **The New Multi-screen World**: Understanding Cross-platform Consumer Behavior [s. L.]: [s. N.], 2012. Disponível em: <<http://bit.ly/1gVzizF>>. Acesso em: 06 mar. 2015.
- GUTTMAN, E. Service location protocol: automatic discovery of IP network services. **IEEE Internet Computing**, [s.l.], v. 3, n. 4, p.71-80, 1999.
- LEVIN, M. An Ecosystem of Connected Devices. In: **Designing Multi-Device Experiences: An Ecosystem Approach to User Experiences Across Devices**. Sebastopol: O'reilly, 2014. p. 1-19.
- MESHKOVA, E. *et al.* A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. **Computer Networks**, [s.l.], v. 52, n. 11, p.2097-2128, 2008.
- UPNP. **UPnP Device Architecture 1.1**. [s.l.:s.n], 2008. Disponível em: <<http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf>>. Acesso em: 23 fev. 2015.
- CHANDRA, Surendar; ROWE, Lawrence A.. DisplayCast: a high performance screen sharing system for intranets. In: ACM International Conference On Multimedia, 20., 2012, Nara, Japan. **Proceedings** . New York: Acm, 2012. p. 1453 – 1456
- CORTEZ, James; SHAMMA, David A.; CAI, Linghan. Device Communication: A Multi-modal Communication Platform for Internet Connected Televisions. In: European Conference on Interactive Tv and Video, 10., 2012, Berlin. **Proceedings** . New York: Acm, 2012. p. 19 – 26.
- PEDROSA, Diogo et al. A Multimodal Interaction Component for Digital Television. In: ACM Symposium on Applied Computing, 2011, Taichung, Taiwan. **Proceedings**. New York: Acm, 2011. p. 1253 – 1258.
- HORNG, Mong-fong et al. Intelligent Self-Management Home Multimedia Service System. In: International Conference on Innovative Computing, Information and Control, 2007 (*ICICIC '07*). **Proceedings** . 2007. p. 199 - 202.