Guilherme Mussi Toschi

Extensão do padrão UPnP para a integração de sistemas de automação residencial

São Paulo

2016

Guilherme Mussi Toschi

Extensão do padrão UPnP para a integração de sistemas de automação residencial

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção de título de Mestre em Ciências

São Paulo

2016

Guilherme Mussi Toschi

Extensão do padrão UPnP para a integração de sistemas de automação residencial

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção de título de Mestre em Ciências

Área de Concentração: Engenharia de Computação

Orientador: Prof. Livre Docente Carlos Eduardo Cugnasca

São Paulo

2016

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, ... de .....de 20.....

Assinatura do autor

Assinatura do orientador

Catalogação-na-publicação

Toschi, Guilherme Mussi
Extensão do padrão UPnP para a integração de sistemas de automação residencial. / G. M. Toschi, C. E. Cugnasca -- versão corr. -- São Paulo, 2016.
110 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.UPnP 2.Automação residencial I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t. III.Cugnasca, Carlos Eduardo

Dedico este trabalho àqueles que sonham em um dia viver em um mundo totalmente automatizado no qual nossas residências serão autônomas e autossuficientes.

AGRADECIMENTOS

Ao professor Carlos Cugnasca pela orientação e pelo constante estímulo transmitido durante o trabalho. À minha namorada que sempre me apoiou, ajudou e foi muito compreensiva comigo durante esse período. Aos amigos e familiares que ajudaram durante essa fase da minha vida, e também a todos que colaboraram direta ou indiretamente na execução deste trabalho.

“Não desista de um sonho por causa do tempo que levará para realizá-lo. O tempo passará de todo jeito.”
(Earl Nightingale)

RESUMO

O presente trabalho discute a compatibilidade e integração entre sistemas e dispositivos de automação residencial, propondo formas de melhorá-la. Essa integração tende a se tornar uma tarefa complexa devido à grande variedade de padrões e tecnologias de integração adotados na automação residencial. O presente trabalho propõe uma extensão do padrão *Universal Plug and Play* (UPnP) e a utilização de uma arquitetura modular com duas camadas, afim de adaptá-lo à integração dos subsistemas de automação residencial. Esse padrão estendido é, então, utilizado na camada superior, para o controle e integração entre os subsistemas. Já na camada inferior, cada subsistema utiliza a tecnologia de comunicação mais adequada para controlar seus dispositivos, e possui uma interface UPnP para se comunicar com outros subsistemas e permitir seu controle pelo usuário. Dessa forma os subsistemas tornam-se módulos do sistema de automação da residência. Essa proposta permite que o usuário compre e substitua facilmente subsistemas de fabricantes distintos, de forma a integrá-los, resultando em um sistema de automação residencial flexível e independente de fabricante. Para testar a extensão proposta, um caso de uso de um subsistema de iluminação foi criado. A partir deste, foram realizadas simulações computacionais. Os resultados destas foram apresentados e analisados, verificando-se o atendimento aos requisitos do sistema e se as características desejadas foram alcançadas, tais como, a característica *plug and play* de subsistemas, o aumento da flexibilidade e a modularização do sistema, para facilitar a compra e manutenção de sistemas de automação residencial, gerando o potencial para fomentar a maior adoção de sistemas de automação residencial. No entanto, a extensão proposta também resulta no aumento da complexidade do cliente UPnP que a utiliza para interagir com o sistema, o que pode dificultar a adoção de sistemas de automação residencial no futuro. Por fim, sugestões de continuidade e perspectivas futuras foram apresentadas.

Palavras-chave: UPnP. Automação residencial. Residência inteligente.

ABSTRACT

This work discuss the compatibility an integration between systems and devices for home automation, proposing ways to improve it. This integration tends to become a complex task due to the wide variety of integration standards and technologies adopted in home automation. This paper proposes an extension of the standard Universal Plug and Play (UPnP) and the use of a modular architecture with two layers in order to adapt it to the integration of home automation subsystems. This extended standard is then used in the upper layer for the control and integration of the subsystems. In the bottom layer, each subsystem uses the most appropriate communication technology to control their devices, it has an UPnP interface to communicate with other subsystems and allow user control. Thus subsystems become modules of the home automation system. This proposal allows the user to easily purchase and replace subsystems from different manufacturers, in order to integrate them transparently, resulting in a flexible home automation system and independent of manufacturer. To test the proposed extension, a use case of an illumination subsystem was created and software simulations were carried out. The simulation results were presented and analyzed, verifying the compliance with the system requirements and whether the desired characteristics have been achieved, such as the subsystem plug and play feature, the modularity and flexibility enhancement of the system, to facilitate the purchase and maintenance of the home automation system, generating the potential to foster greater adoption of home automation systems. However, the proposed extension also results in increased complexity of the UPnP client that uses it to interact with the system, which can hinder the adoption of home automation systems in the future. Finally, future suggestions and perspectives were presented.

Keywords: UPnP. Home automation. Smart home.

LISTA DE QUADROS

Quadro 1 - Exemplos de padrões de infraestrutura residencial *13*

Quadro 2 - Comparação qualitativa *49*

Quadro 3 - Análise SWOT do UPnP estendido *85*

Quadro 4 - Análise SWOT padrões atuais *86*

LISTA DE FIGURAS

Figura 1 - Diagrama da taxa de transferência de dados por custo 23

Figura 2 - Diagrama de classificação dos meios de transmissão em HANs 25

Figura 3 - Exemplos de redes residenciais 28

Figura 4 - Protocolo multicamada UPnP 29

Figura 5 - Protocolo em camadas do ZigBee 37

Figura 6 - Cabeçalho do protocolo X-10 39

Figura 7 - Protocolo multicamadas do Home Plug and Play 40

Figura 8 - DomoNet TechManager 41

Figura 9 - Arquitetura DomoNet 42

Figura 10 - Protocolo multicamadas do HYDRA 45

Figura 11 - Possível arquitetura futura 57

Figura 12 - Estrutura de descrição de dispositivos UPnP 65

Figura 13 - Modelo estendido para sistemas multidispositivos 66

Figura 14 - Arquitetura proposta para o modelo estendido 68

Figura 15 - Dispositivo exemplo 73

Figura 16 - Cliente UPnP da ferramenta 73

Figura 17 - Telas do Service Author para o serviço de dimerização 75

Figura 18 - Telas do Service Author para o serviço de liga/desliga 75

Figura 19 - Serviço do subsistema de iluminação 76

Figura 20 - Descrição do sistema de iluminação 77

Figura 21 - Geração da estrutura de código UPnP 78

Figura 22 - Projeto do Visual Studio 79

Figura 23 - Interface do sistema de iluminação 80

Figura 24 - Ação de uma lâmpada do sistema de iluminação 80

Sumário

1. INTRODUÇÃO 12
1.1 MOTIVAÇÃO E JUSTIFICATIVAS 12
1.1.1 Sobre a Automação Residencial 12
1.1.2 Dispersão das Soluções de Automação 14
1.1.3 Arquitetura 17
1.2 OBJETIVO 18
1.3 CONTEÚDO E ORGANIZAÇÃO 19
2. REVISÃO DA LITERATURA 20
2.1 INTRODUÇÃO 20
2.2 HOME AREA NETWORK (HAN): ELEMENTOS E CONCEITOS 22
2.2.1 Princípios 22
2.2.2 Definições 25
2.3 PADRÕES, ARQUITETURAS E INICIATIVAS 27
2.3.1 Padrões 28
2.3.2 Arquiteturas 40
2.3.3 Iniciativas 44
2.3.4 Resumo das tecnologias 47
2.4 DESAFIOS EM ABERTO E DIRECIONAMENTOS FUTUROS 53
2.4.1 Questões em discussão 54
2.4.2 Futuro 55
2.5 FECHAMENTO 58
3. MATERIAIS E MÉTODOS 60
3.1 ESCOLHA DO PADRÃO 60
3.2 ELABORAÇÃO DO MODELO ESTENDIDO 60
3.3 DEFINIÇÃO DA ARQUITETURA 61
3.4 SIMULAÇÃO COMPUTACIONAL DO MODELO ESTENDIDO 62
3.5 AVALIAÇÃO DOS RESULTADOS E DISCUSSÕES 63
3.6 FECHAMENTO 63
4. PROPOSTA DE EXTENSÃO DO MODELO E DEFINIÇÃO DA ARQUITETURA 64
4.1 ESCOLHA DO UPNP 64

4.2 O MODELO UPNP 64
4.3 PROPOSTA DE EXTENSÃO DO MODELO UPNP..... 65
4.4 PROPOSTA DA ARQUITETURA DO SISTEMA DE AUTOMAÇÃO 67
4.5 ANÁLISE DA PROPOSTA..... 69
4.6 FECHAMENTO..... 71
5. SIMULAÇÃO COMPUTACIONAL DO MODELO PROPOSTO..... 72
5.1 FERRAMENTAS DE IMPLEMENTAÇÃO E SIMULAÇÃO 72
5.2 CASO DE USO PARA VALIDAÇÃO DO MODELO PROPOSTO..... 74
5.3 IMPLEMENTAÇÃO DO CASO DE USO..... 74
5.3.1 Criação do modelo do subsistema de iluminação..... 74
5.3.2 Código C do subsistema de iluminação..... 78
5.4 APRESENTAÇÃO DAS SIMULAÇÕES REALIZADAS 80
5.5 FECHAMENTO..... 81
6. RESULTADOS E DISCUSSÕES..... 82
6.1 ANÁLISE DOS RESULTADOS OBTIDOS..... 82
6.2 COMPARATIVOS COM AS SOLUÇÕES JÁ EXISTENTES..... 85
6.3 DISCUSSÕES 86
7. CONCLUSÕES..... 88
7.1 PERSPECTIVAS DE CONTINUIDADE DO TRABALHO..... 89
7.2 FECHAMENTO..... 90
8. REFERÊNCIAS..... 91
APÊNDICE A..... 103

1. INTRODUÇÃO

Este capítulo apresenta a motivação e justificativas para realização deste trabalho, relacionado com a automação residencial, evidenciando deficiências encontradas nessa área, que justifiquem os objetivos desta pesquisa.

1.1 MOTIVAÇÃO E JUSTIFICATIVAS

Este item apresenta a motivação e as justificativas para a escolha da automação residencial como foco deste trabalho.

1.1.1 Sobre a Automação Residencial

A automação residencial vem ganhando maior destaque desde o início dos anos 2000. Nesse período várias entidades desenvolveram padrões e especificações destinadas à implementação de uma infraestrutura residencial, como apresenta Wacks (2002) em seus estudos, nos quais questiona se essas especificações têm sido bem-sucedidas.

Na década de 1980 começou-se a pesquisar e definir padrões, que na década seguinte começaram a ser introduzidos no mercado. Foi nesta época, com a popularização da Internet, que as empresas começaram a desenvolver suas próprias especificações para redes residenciais, baseadas no padrão TCP/IP. Um exemplo relevante de empresa de destaque nessa área é a Echelon Corporation (ECHELON, 2012), que desenvolveu um padrão fechado e proprietário para automação predial, o LonWorks©. Posteriormente, esse padrão se transformou em um padrão aberto, homologado na ISO/IEC 14908 e no ANSI/CEA-709.1-B (BOLZANI, 2004).

Ainda no início dos anos 2000, outras empresas se uniram para formar entidades de padronização e criar padrões de modo colaborativo, sendo que algumas dessas organizações continuam ativas buscando novos participantes que ajudem na evolução de seu padrão e que o adotem em seus produtos e sistemas (BOLZANI, 2004). Um exemplo é a entidade UPnP Fórum, criada em outubro de 1999, que desenvolveu o padrão *Universal Plug and Play* (UPnP), publicado como

parte da norma internacional ISO/IEC 29341 em dezembro de 2008 (UPNP FORUM, 2014). O Quadro 1 apresenta alguns exemplos de padrões existentes.

Quadro 1 - Exemplos de padrões de infraestrutura residencial

Exemplo de Padrões
DOCSIS: Cable modem
EIA-600: CEBus
EIA-721:Generic CAL
EIA-776: CEBus-EIB-router
EIA-844: XML for Generic CAL
TIA-570A: Residential wiring
Exemplo de Protocolos oferecidos por Empresas e Consórcios
AV/C and HAVi
Bluetooth
HomeCNA
Home Plug & Play
HomePlug
HomePNA
HomeRF
JINI
OSGi
Universal Plug and Play/SCP
Exemplo de Protocolos Proprietários
HomeCast
HomeConnex
Home Wireless Network
Jack Rabbit
MediaWire Home Network
RadioLAN
ShareWave
Symphony

Fonte: Wacks (2002)

Na automação residencial existe uma grande variedade de tecnologias e protocolos que são utilizados em padrões abertos e proprietários (WACKS, 2002). Esse mesmo cenário se aplica à automação predial (WANG et al., 2007), e ambos sofrem com a incompatibilidade e a falta de integração entre os dispositivos e subsistemas de automação criados por essa variedade de soluções heterogêneas. Porém o cenário da automação residencial está segmentado e é neste aspecto que a integração seria mais benéfica.

A falta de integração entre as diferentes tecnologias, a heterogeneidade na comunicação de controle dos dispositivos e a dificuldade de padronização do

mercado resultam em problemas relacionados à interoperabilidade entre as diferentes soluções existentes (WACKS, 2002; NGO,2007). Assim, não se consegue utilizar estas soluções de modo cooperativo operando simultaneamente de forma integrada, mas somente como soluções desconexas operando em paralelo.

O resultado dessa situação são usuários dependentes de uma única solução e fornecedor, que não podem combinar os melhores módulos e dispositivos que utilizam tecnologias distintas para compor e personalizar o seu sistema de automação residencial, de modo a satisfazer seus requisitos e necessidades. Dessa forma, o desenvolvimento desintegrado, sem escalabilidade e interoperabilidade das novas soluções se deve à falta de padronização na automação residencial (MIORI et al., 2006).

Com a compatibilidade entre os diversos módulos e dispositivos, que utilizam diferentes tecnologias de automação residencial, seria possível integrar soluções que seguem padrões distintos de diversos fabricantes, ampliando as possibilidades de escolha do usuário. Ao criar essa liberdade de integração entre produtos de diferentes fabricantes, espera-se obter um melhor e mais rápido desenvolvimento e adoção de sistemas de automação residencial.

A partir dessa visão, pretende-se fazer uma investigação dos diferentes protocolos de comunicação de automação residencial e das soluções existentes de compatibilidade.

1.1.2 Dispersão das Soluções de Automação

A arquitetura tradicional de *smart homes* é centralizada em torno de um servidor, resultando em dispositivos que não funcionam de forma independentes (NGO, 2007). Nessa arquitetura, os componentes da rede residencial são controlados por um nó central que se torna um *home gateway* (NGO, 2007), isto é, o usuário controla os dispositivos através deste. Porém, estes dispositivos estão se tornando interdependentes e trocando informações entre si, pois que essa arquitetura não está preparada para atender às novas necessidades de interconexão entre os dispositivos (NGO, 2007; WACKS, 2002).

Outra abordagem foi a utilização de uma camada intermediária de integração que controla a interação e comunicação entre esses dispositivos. Essa camada

utiliza a tecnologia de comunicação que é mais adequada para o tipo de dispositivo gerenciado por esta solução. Dessa forma, uma única solução não consegue fornecer o grau de interoperabilidade necessário para integrar diferentes tipos de aplicações que podem ser encontradas em uma residência, e por isso, é improvável que um destes se torne dominante para a computação pervasiva em uma residência (MIORI et al., 2006).

Seguindo essa segunda abordagem, muitas tecnologias de comunicação são adotadas com propósitos variados por fabricantes de sistemas de automação residencial. Essas diferentes tecnologias utilizadas são independentes, criando uma barreira para a integração entre elas e assim esses sistemas de controle da residência não são compatíveis, não podendo ser utilizados cooperativamente, o que dificulta a interoperabilidade entre as soluções já existentes no mercado. Dessa forma, o usuário fica dependente de uma única tecnologia, não podendo utilizar os melhores módulos de tecnologias distintas.

Para utilizar essas tecnologias em conjunto, é necessária uma conexão dedicada e especializada entre cada par distinto de tecnologias (NGO, 2007; MIORI et al., 2006). Segundo Ngo (2007), as soluções de compatibilidade e integração existentes são pontes, isto é, conexões entre duas tecnologias distintas, de forma a limitar suas funcionalidades (NGO, 2007). Dessa forma, cria-se a necessidade de fazer uma conversão da linguagem de controle, que utiliza um protocolo de comunicação dedicado, em outro protocolo com propósito diferente (MIORI et al., 2006). Essas soluções afetam a funcionalidade de cada tecnologia, limitando seus controles intrínsecos ao realizar essa tradução.

Miori; Russo e Aliberti (2010) apresentam outra linha de pesquisa para essa área, que é a utilização de serviços *web* para integrar essas tecnologias distintas e, em seu artigo descrevem o objetivo comum às várias pesquisas de integrar tecnologias heterogêneas sem a perda de suas funcionalidades:

> *"… a computing framework based on open communication standards, capable of abstracting the peculiarities of underlying heterogeneous technologies, and letting them co-exist and interwork, without eliminating their differences."*

Porém a solução encontrada foi uma ponte múltipla, na qual cada tecnologia tem o seu tradutor para um serviço *web*. Todos os serviços têm a sua linguagem, obedecendo a um padrão de controle abstrato, e esse padrão deve ser capaz de

sintetizar todas as possíveis comunicações de leitura e atuação de todos os possíveis dispositivos que se conectem a uma das tecnologias de comunicação suportadas pela solução.

Essa abordagem pode resolver o problema de sobrecarregar os dispositivos com um protocolo mais generalista e abstrato, deixando-os com uma comunicação dedicada e mais adequada ao propósito de cada dispositivo. Mas o mapeamento entre a linguagem generalista utilizada nos serviços e a linguagem dedicada de cada dispositivo é uma tarefa complexa, e demanda a atualização do nó central quando houver mudanças e adições nos protocolos e linguagens utilizadas pelos dispositivos. Esse problema pode ser solucionado pela utilização, por todos os dispositivos, de uma única linguagem padrão de automação que atuaria independente dos protocolos de tecnologias de comunicação (MIORI et al., 2006; MIORI; RUSSO; ALIBERTI, 2010). Porém, a proposição desse protocolo e sua adoção pelos fabricantes de dispositivos inteligentes não é uma tarefa trivial.

Outra solução que poderia ser proposta foi a criação de um protocolo de alto nível, que se torne um padrão para todos os dispositivos de automação. Ele definiria as mensagens permitidas a trafegar por um meio físico, através de um protocolo de baixo nível (de instrumentação), constituindo-se em uma primeira alternativa para a solução desse problema. Esse protocolo abstrato deve ser independente do meio físico para se tornar generalista e poder ser adotado universalmente.

Mas essa solução acarretaria em processar um protocolo de alto nível em todos os dispositivos, e assim, cada dispositivo, por mais simples que fosse, deveria possuir um hardware poderoso o suficiente para poder suportar esse protocolo padronizado. Dessa forma, esse protocolo poderia prover todos os possíveis controles de todos os dispositivos da residência, tanto os existentes como os que sejam criados posteriormente.

Essa solução gerou a questão sobre em qual parte da arquitetura deveria ser feita essa abstração para aumentar o nível da linguagem utilizada, e além disso, levantou-se a questão sobre qual arquitetura seria capaz de sustentar essa abstração, e se esta poderia ser utilizada como metodologia de desenvolvimento de modo adequado e flexível para se adaptar a novos dispositivos, independentemente de fabricantes e de tecnologias de comunicação específicas utilizadas.

1.1.3 Arquitetura

O estudo de uma arquitetura adequada pode possibilitar um desenvolvimento contínuo e uma integração entre subsistemas distintos e heterogêneos. Essa arquitetura deve ser concebida de modo a promover a livre concorrência, através da possibilidade de integração de produtos padrões de diferentes fabricantes, diminuindo a dependência tecnológica, uma vez que esses módulos podem ser substituídos por novos, com novas tecnologias, quando necessário.

Segundo Gama; Touseau e Donsez, (2012), as soluções baseadas na Arquitetura Orientada a Serviços (*Service Oriented Architecture* - SOA*)* que utilizam os princípios da Computação Orientada a Serviços (*Service Oriented Computing* - SOC), podem atender aos requisitos na integração de dispositivos de automação residencial. Conceitos de engenharia de software orientada a serviços podem ser utilizados no desenvolvimento de sistemas de automação residencial, no qual um subsistema de automação se torna um serviço, e assim, modularizando-se a solução. Dessa forma, a solução pode ser entregue gradualmente ao usuário final em partes, funcionais autossuficientes, e depois integrada em um sistema composto por vários desses módulos.

A arquitetura física dos dispositivos em uma residência pode ser considerada distribuída e heterogênea devido à grande diversidade de tecnologias utilizadas nesses dispositivos e à sua distribuição física dentro da própria residência, somando-se a isso o fato de haver uma grande disponibilidade e facilidade de acesso à infraestrutura de comunicação residencial, como a Wi-Fi, cria-se o objetivo de conectar todos esses dispositivos.

Existem sistemas de automação residencial que utilizam diferentes métodos e tecnologias que as tornam incompatíveis entre si. Essa deficiência gera a necessidade de se criar uma solução de integração para sistemas de automação residencial.

Esse cenário da automação residencial está diretamente relacionado com a Internet das Coisas (*Internet-Of-Things* - IoT). Miorandi et al. (2012) descrevem IoT da seguinte maneira:

> *"Internet-of-Things envisions a future in which digital and physical entities can be linked, by means of appropriate information and communication technologies, to enable a whole new class of applications and services."*

Os conceitos associados a IoT cobrem vários aspectos relacionados à extensão da Internet ao mundo físico, através da integração de dispositivos espalhados e distribuídos espacialmente, com a capacidade de se identificar, sentir e atuar sobre o ambiente (MIORANDI et al., 2012). Segundo Atzori; Iera e Morabito (2010), o fator mais promissor desse novo paradigma é a integração de diversas tecnologias e soluções de comunicação.

A partir desse cenário mais abrangente do IoT, pode-se considerar a automação residencial como uma simplificação e um caso de uso do IoT, e assim pode aproveitar-se de técnicas, tecnologias e modelos de integração desenvolvidos e implementados para esta.

Este cenário, apesar de reduzido, mostra-se bem intenso. Em 13 de janeiro de 2014, a empresa *Google Inc.* anunciou um acordo para comprar a empresa *Nest Labs Inc.* por 3,2 bilhões de dólares (TILLEY, 2014), cuja missão é reinventar dispositivos residenciais, e desde seu lançamento, em 2011, vem experimentando aumentos em suas vendas (GOOGLE PRESS, 2014). Esse foi um grande passo da Google para entrar no mercado de automação residencial, mas não foi o primeiro esforço para atingir esse objetivo. Porém, no final de 2013, a Google já lançou um serviço de monitoramento do consumo de energia elétrica (FEHRENBACHER, 2013). Essa empresa não é a única com interesse nesse mercado. O *chief executive officer* (CEO) da empresa Cisco, John Chambers, afirmou, no começo de 2014, que o mercado de IoT pode gerar 19 trilhões de dólares por permitir que dispositivos e aplicações trabalhem juntos e criem novos serviços fora da Internet (GUGLIELMO, 2014).

1.2 OBJETIVO

Este trabalho tem como objetivo propor uma a extensão do padrão do Universal Plug and Play (UPnP), utilizado na camada de controle e integração da arquitetura de sistemas de automação residencial, de forma a contribuir com a mitigação do problema de incompatibilidade e dificuldade de integração entre os subsistemas de automação residencial.

1.3 CONTEÚDO E ORGANIZAÇÃO

Este trabalho de pesquisa é dividido em sete capítulos. No Capítulo 1 a motivação, justificativas e o objetivo da pesquisa são apresentados. No Capítulo 2 a revisão de literatura associada à área de pesquisa é apresentada, demonstrando o seu estado da arte atual, com ênfase no levantamento das principais tecnologias e padrões de automação residencial. No Capítulo 3 os materiais e métodos empregados na pesquisa são apresentados, definindo-se a sequência de atividades: (i) escolha do padrão, (ii) elaboração do modelo estendido, (ii) definição da arquitetura e (iv) simulação computacional do modelo proposto.

No Capítulo 4 a extensão do modelo UPnP e a arquitetura da solução são propostos e discutidos. No Capítulo 5 as ferramentas de implementação e simulação utilizadas são apresentadas, o caso de uso definido, e as simulações do modelo estendido são descritas. O Capítulo 6 apresenta os resultados e as discussões da extensão proposta para o modelo UPnP. Este modelo é analisado qualitativamente da mesma forma que as outras tecnologias e padrões para demostrar suas vantagens e desvantagens. Finalizando, o Capítulo 7 apresenta as conclusões do trabalho e perspectivas de sua continuidade.

Após as Referências, um apêndice foi incluído com o código-fonte da classe principal do subsistema de iluminação utilizado na simulação.

2. REVISÃO DA LITERATURA

As Redes de Automação Residencial oferecem uma oportunidade de construção de sistemas de *smart homes*. Nesse cenário, as redes Máquina-a-Máquina (M2M) vêm se constituindo como um método eficiente de prover comunicação automatizada entre os dispositivos distribuídos (KIM; YOUM, 2013; KIM et al., 2014). Em um esforço de apresentar as tecnologias usadas em M2M no Ambiente de Residencial, este capítulo apresenta os elementos e as definições de *Home Area Network* (HAN) e revisa padrões, arquiteturas e iniciativas criadas e desenvolvidas para permitir a comunicação e integração M2M em diferentes ambientes, especialmente, no domínio de *smart homes* (KIM; YOUM, 2013; KIM et al., 2014). Uma comparação entre elas foi elaborada e apresentada para identificar as diferenças e tendências para o futuro desse mercado.

2.1 INTRODUÇÃO

A Automação Residencial evoluiu para algo além da interconexão entre dispositivos autônomos, pois vem possibilitando a concepção de sistemas e processos que estão se tornando cada vez mais complexos, poderosos e capazes de interagir com os moradores da residência. De acordo com Lyytinen e Yoo (2012), os computadores estarão integrados nas interações naturais dos usuários com os ambientes, tanto nas físicas como nas sociais. Isso significa que se está próximo de se ter uma relação social com um sistema autônomo de forma transparente. Essa proposta é similar à que Mark Weiser vislumbrou em 1993, na qual computadores permitiriam a integração de objetos simples como etiquetas de roupa, ar condicionado, interruptores de luz, óculos, entre outros, de forma invisível na vida dos usuários através da computação móvel e pervasiva que se baseia nos princípios de diversidade, conectividade e descentralização (FERREIRA; CANEDO; SOUSA JUNIOR, 2013; WEISER, 1993, 1999).

Este fenômeno é parcialmente explicado pela constante queda no custo de produção do hardware, pelo avanço nas pesquisas científicas e pelo crescimento dos investimentos no setor. A expectativa para 2025 é que o impacto econômico da automação do *knowledge work* seja de 5 a 7 trilhões de dólares (MANYIKA et al.,

2013). O crescente investimento no setor de automação tem diversificado o alcance das suas aplicações. Se quando a automação surgiu, ela era restrita a domínios com alto valor agregado tais como, indústria (DAHM; MATHUR, 1990; ÅSTRÖM, 1985), militarismo (LEINER et al., 2009; MCQUILLAN; RICHER; ROSEN, 1980), ou corrida espacial (KUIPERS; BYUN, 1991), atualmente está cada vez mais próxima do dia a dia das pessoas, por exemplo, em aplicações de controle de acesso (XU et al., 2012), sistemas de estacionamento inteligentes (GENG; CASSANDRAS, 2013) e aplicativos para *smartphones* (BOULOS et al., 2011). O foco dessas aplicações é fornecer o bem-estar, aumentar a qualidade de vida e melhorar a saúde e serviços de segurança.

Para atingir esse objetivo, a automação incluiu recursos de processamento de imagens e é capaz de utilizar informações combinadas de vários sensores e acionadores (VARCHOLA; DRUTAROVSKÝ, 2007). Este ambiente, que possui um grande volume de dados, contribui para a criação de novos conceitos e novos campos de pesquisa, como Internet das Coisas (ATZORI; IERA; MORABITO, 2010), *Big Data* (MCAFEE, 2012), Inteligência Artificial, entre outros (COOK; AUGUSTO; JAKKULA, 2009). Estes conceitos têm evoluído para novos domínios de aplicação, entre eles, cidades inteligentes, agricultura, transporte, saúde inteligente, etc. A ideia principal é a existência de processos capazes de atender às demandas específicas de outros processos ou de pessoas (COOK; DAS, 2007).

Desta forma, a *smart home* é um domínio de aplicações bastante familiar para as pessoas, e é capaz de fundir todos esses novos conceitos (INTILLE, 2002, 2006). Assim, esta pesquisa retrata o estado da arte atual do cenário da automação residencial e revisa os métodos a integração de redes.

O domínio da automação residencial é caracterizado por uma infraestrutura que permite uma rede inteligente de dispositivos e aparelhos que utilizam várias tecnologias de comunicação, com e sem fio, para fornecer uma integração de qualidade com subsistemas residenciais, e de uso fácil pelos seus habitantes enquanto cria um ambiente personalizado e seguro para a residência (KUMAR et al., 2014). A projeção de crescimento do mercado de automação residencial é expressiva, evoluindo de 40 milhões de dólares em 2012, para 26 bilhões de dólares em 2019 (FAGNANT; KOCKELMAN, 2015). Essa projeção vem atraindo uma série de empresas, como General Eletric, Cisco, Google e Microsoft, que estão unindo

esforços para o desenvolvimento de novos padrões, protocolos e otimização de funções e métodos dos dispositivos de automação residencial (GUGLIELMO, 2014; TILLEY, 2014; GOOGLE PRESS, 2014).

No item 2.2 deste capítulo serão descritos os elementos e conceitos da *Home Area Network*. Já o item 2.3 compara padrões, arquiteturas e iniciativas do cenário da automação residencial. Finalmente, nos itens 2.4 e 2.5, são apresentados desafios em aberto e perspectivas de evolução para esse mercado.

2.2 HOME AREA NETWORK (HAN): ELEMENTOS E CONCEITOS

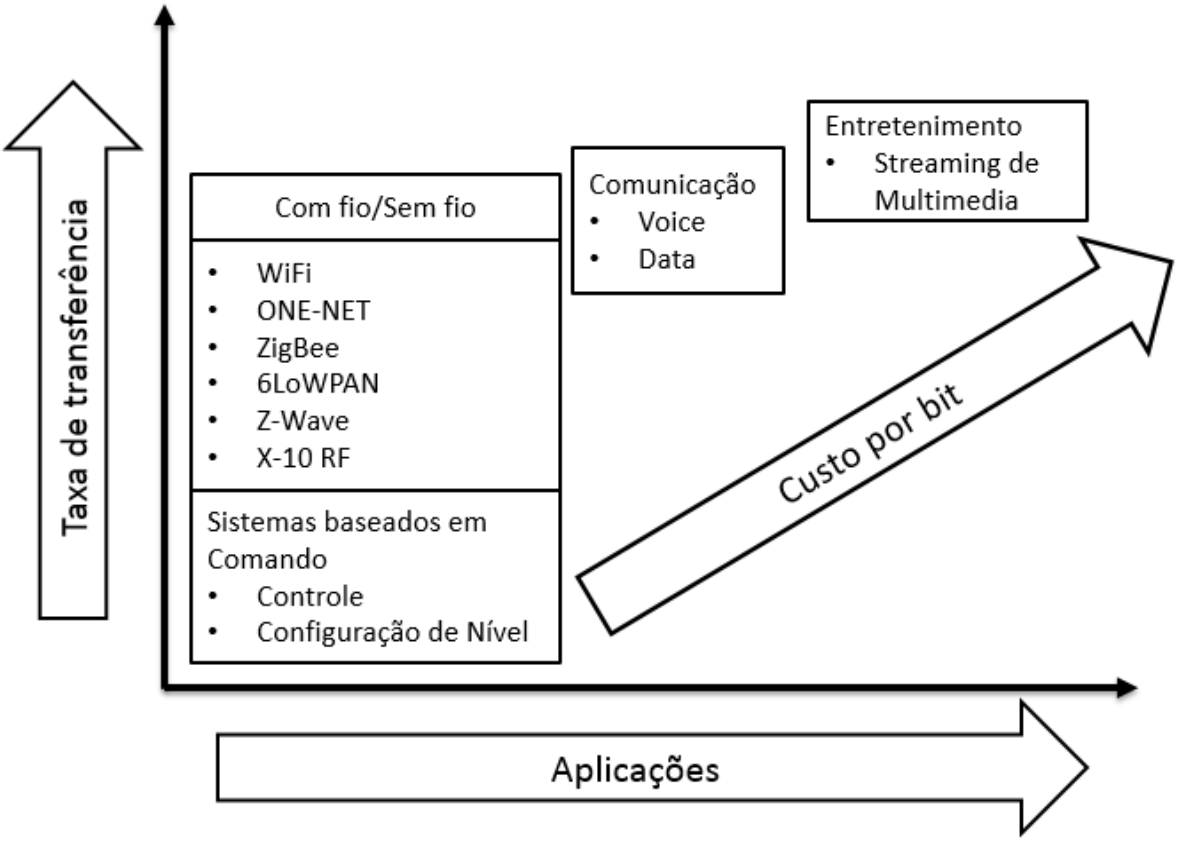
Smart home e automação residencial vêm sendo estudadas desde o Século XX, e o crescimento deste mercado se deve ao aumento do interesse e da demanda dos consumidores. Na primeira década do Século XXI passou a receber ainda mais atenção (BALTA-OZKAN et al., 2013).

2.2.1 Princípios

O termo HAN foi definido por Rose (2001) como uma rede para interconectar produtos e sistemas eletrônicos da residência, permitindo aos moradores acessá-los e controlá-los remotamente, e disponibilizando qualquer conteúdo como música, vídeos e outros dados. Além disso, Kailas, Cecchi e Mukherjee (2012) descrevem HAN como uma rede para conectar dispositivos capazes de enviar e receber sinais de outros dispositivos e aplicativos. HAN é, assim, um conceito que possibilita a evolução da *smart home* e de outras aplicações relacionadas, como sistemas de gestão de energia residencial (*home energy management*, HEM). Essa rede que liga dispositivos domésticos é a base do conceito de *smart home* (BALTA-OZKAN et al., 2013).

Uma HAN pode abranger diferentes redes de tecnologias distintas para melhor atender os requisitos do sistema. Dependendo do tipo de rede, a infraestrutura pode ser mais especializada e ter maior velocidade de dados, ou ser mais flexível e possibilitar uma interação mais ampla de aplicações em relação a outras sub-redes de outros subsistemas. Essas características impactam no custo da infraestrutura para essa rede. A Figura 1 exemplifica essa relação do custo.

Figura 1 - Diagrama da taxa de transferência de dados por custo



Fonte: adaptado de Kailas; Cecchi e Mukherjee (2012)

2.2.1.1 Internet Protocol (IP)

O IP é uma camada da pilha de protocolos TCP/IP e forma o alicerce a partir do qual a Internet é construída. De acordo com Kailas; Cecchi e Mukherjee (2012), devido à simplicidade de interoperabilidade do IP, a sua grande adoção e de iniciativas (para a criação de interfaces que exijam menos recursos computacionais), o IP é considerado essencial para o sucesso do HANs.

No começo dos anos 2000, passou a ser comum se encontrar residências que utilizam redes baseada no IP, de forma que se pode considerar que as HANs já estão, portanto, presentes em muitas residências. Porém, ainda há a dificuldade associada à integração de muitos dispositivos e componentes baseados em IP, com aqueles não baseados em IP (CHEN et al., 2014). Essa conexão pode ser feita através de um *gateway* multiprotocolo, que faz uma ponte entre redes IP e outros tipos de rede (CHEN et al., 2014). De acordo com Ngo (2007), essa ponte entre

tecnologias é limitada, e sempre que uma nova tecnologia ou rede é criada, esse novo mapeamento precisa ser implementado, criando-se assim uma nova ponte. E assim, essas conexões não podem ser usadas como solução final para a rede doméstica devido à essas limitações e necessidade de manutenção constante (NGO, 2007).

2.2.1.2 HANs com fio

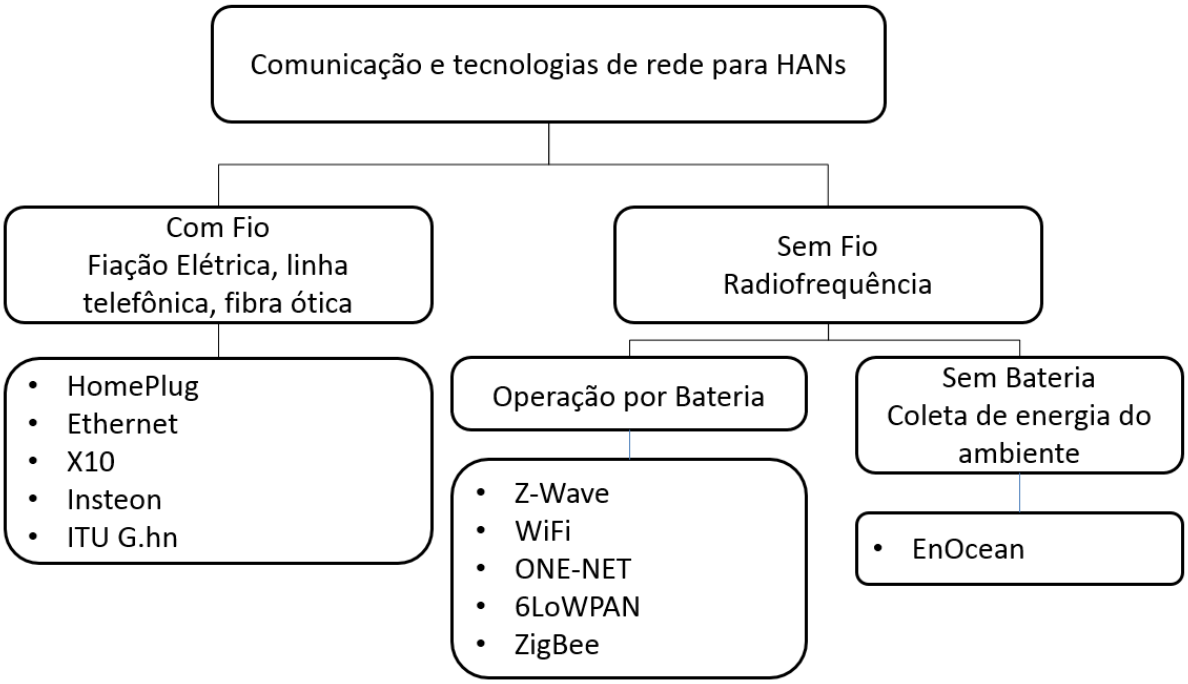
As tecnologias de comunicação com fio foram as primeiras a surgir, e ainda são adotadas por diferentes razões, como o seu baixo custo aparente ou o uso da infraestrutura já existente em uma residência. HANs cabeadas podem utilizar as linhas de energia, linhas de telefone, cabos coaxiais, par trançado ou tecnologias mais recentes, como a fibra óptica, como meio de comunicação.

2.2.1.3 HANs sem fio

Tecnologias de comunicação sem fio vêm sendo pesquisadas e desenvolvidas com mais intensidade nos anos 2000. É esperado que elas exercessem um papel fundamental na expansão do conceito de residência digital (VAXEVANAKIS; ZAHARIADIS; VOGIATZIS, 2003). A sua adoção pelos usuários é vasta, pois interfaces para essa comunicação estão presentes em muitos dos computadores pessoais, console de jogos, dispositivos periféricos e dispositivos domésticos (KAILAS; CECCHI; MUKHERJEE, 2012). HANs sem fio apresentam desde padrões residenciais populares, como Wi-Fi e ZigBee, até novos padrões, como o 6LoWPAN que otimiza o IPv6 para um uso de tecnologias de comunicação de baixo consumo de energia, um padrão da *Internet Engineering Task Force* (IETF) (KAILAS; CECCHI; MUKHERJEE, 2012).

A comparação entre as tecnologias de comunicação sem fio e as cabeadas envolve questões como o consumo de energia, a cobertura de sinal, a sensibilidade a interferências, a segurança e a instalação da infraestrutura (KAILAS; CECCHI; MUKHERJEE, 2012). Em suma, HANs possibilitam diversas alternativas de comunicação e formação de redes, utilizando diferentes tecnologias que especificam as camadas físicas e de rede. A classificação das HANs baseadas no meio de transmissão de dados pode ser vista na Figura 2.

Figura 2 - Diagrama de classificação dos meios de transmissão em HANs



Fonte: adaptado de Kailas; Cecchi e Mukherjee (2012)

A proliferação de dispositivos que utilizam padrões de comunicação distintos dificulta manter características uniformes de desempenho, porém esta é uma característica chave para os serviços e aplicações emergentes no cenário de *smart home* (KUMAR et al., 2014).

2.2.2 Definições

O termo *smart home* pode, em princípio, se referir a qualquer forma de residência, por exemplo, uma residência autossuficiente, um apartamento ou uma unidade em uma habitação social (BALTA-OZKAN et al., 2013). De acordo com (BALTA-OZKAN et al., 2013), uma *smart home* pode ser caracterizada por quatro aspectos: (i) infraestrutura de rede de comunicação; (ii) controle e administração inteligente; (iii) rede de sensores na residência; e (iv) características inteligentes, ou seja, respostas automáticas às entradas humanas e de sensores. Entretanto, a essência das *smart homes* são as tecnologias de informação e comunicação distribuídas pela residência, eletrodomésticos e sistemas inteligentes coletando e recebendo informações para

responder aos usuários e os sistemas inteligentes de automação e administração da residência (WILSON; HARGREAVES; HAUXWELL-BALDWIN, 2015).

Nesse sentido, como identificado por Wilson; Hargreaves, e Hauxwell-Baldwin (2015), a pesquisa sobre *smart homes* pode ser dividida em três visões, (i) Funcional, que considera *smart home* como uma maneira de melhor gerenciar as demandas da vida no dia-a-dia, (ii) Instrumental, com foco em gerenciar e reduzir a demanda de energia em dispositivos da residência para um futuro de baixo carbono, e (iii) Técnica-social, que considera a *smart home* como a próxima onda de desenvolvimento a respeito da digitalização da vida cotidiana.

Na visão funcional, *smart homes* devem ser equipadas com tecnologias avançadas para automação funcional (LERTLAKKHANAKUL; CHOI; KIM, 2008). De acordo com Wood e New-Borough (2007), os sistemas de automação residencial são importantes nas *smart homes* pois contribuem para o conforto, o bem-estar e a alta qualidade de vida dos seus usuários (BUYS et al., 2005; DEMIRIS; HENSEL, 2009).

Entretanto, de acordo com a visão técnico-social, *smart homes* podem ser equipadas com sensores e redes sem fio, como aparelhos domésticos modernos com sensores embarcados, levando a uma arquitetura que atende às necessidades dos usuários (KRONER, 1997). Nesse contexto, o termo *smart home* é definido como: "Uma residência equipada com redes de alta tecnologia, interligando sensores e dispositivos domésticos, eletrodomésticos, e características que podem ser monitoradas remotamente, acessadas ou controladas, e que provêm serviços que respondem as necessidades dos seus moradores" (CHAN et al., 2008; KING, 2003).

Na visão instrumental, os componentes principais da *smart home* são os medidores inteligentes (*smart meters*) e dispositivos que contribuem para o uso racional de energia, constituindo-se uma funcionalidade de gerenciamento de energia. Isso permite o controle pelo usuário e a otimização da programação do uso de eletrodomésticos juntamente com a micro geração de energia (PAETZ; DÜTSCHKE; FICHTNER, 2011). Por exemplo, residências individuais podem ser integradas em *smart grids*, com uma melhora na funcionalidade de gerenciamento de energia para aparelhos, e potenciais ganhos de eficiência com benefícios financeiros e ambientais.

Esse conceito pode ser estendido para criar uma cidade inteligente, com diferentes combinações de macro infraestrutura de automação e integração de sistemas de cidades. Em adição a isso, a integração de uma *smart home* em uma cidade inteligente cria campos de estudo novos.

A automação proporcionada pelas *smart homes* pode trazer economia de energia e água, integrando a coprodução de energia com a rede de distribuição, e pode ser considerada o ponto final do sistema de energia inteligente (WILSON; HARGREAVES; HAUXWELL-BALDWIN, 2015). Porém, também existem custos implícitos a considerar, como a instalação e a manutenção, que podem se tornar bastante complexas e custosas devido ao grande número de fornecedores que devem ser integrados no mesmo sistema (ROSE, 2001).

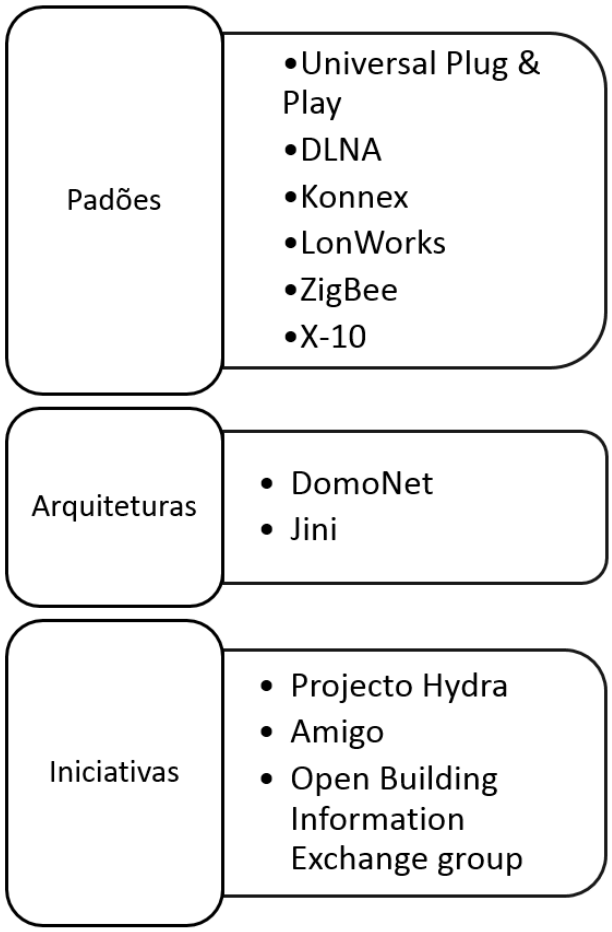
Uma forma de reduzir esses custos é por meio da adoção de padrões. Já existem diversos padrões nessa área, mas poucos estão realmente sendo utilizados pelo mercado. A adoção de um padrão, embora possa trazer diversos benefícios, pode não ser um processo rápido, e, além disso, grandes fornecedores tendem por optar por padrões próprios e fechados (ROSE, 2001).

2.3 PADRÕES, ARQUITETURAS E INICIATIVAS

Esse item apresenta uma revisão das tecnologias utilizadas no M2M e na integração de ambientes residenciais. A Figura 3 apresenta exemplos de tecnologias de redes residenciais. Recentemente, novos protocolos, padrões e arquiteturas estão sendo propostos e desenvolvidos para permitir a comunicação M2M e a integração de vários diferentes ambientes, sendo que alguns são bastante específicos para uma categoria de aplicação, enquanto outros são mais flexíveis e estão sendo usados em ambientes diferentes daqueles para o qual foram concebidos.

Ao analisar algumas dessas tecnologias, pode-se ter uma visão ampla desse campo e identificar tendências e diretrizes para o futuro.

Figura 3 - Exemplos de redes residenciais



Fonte: autor

Na Figura 3, três grupos são apresentados, o grupo Padrões representa os padrões de tecnologias de comunicação utilizadas na automação residencial, o grupo Arquitetura representa padrões de arquiteturas de para automação residencial, enquanto o grupo Iniciativas representa projetos de padronização de automação residencial que provem soluções completas.

2.3.1 Padrões

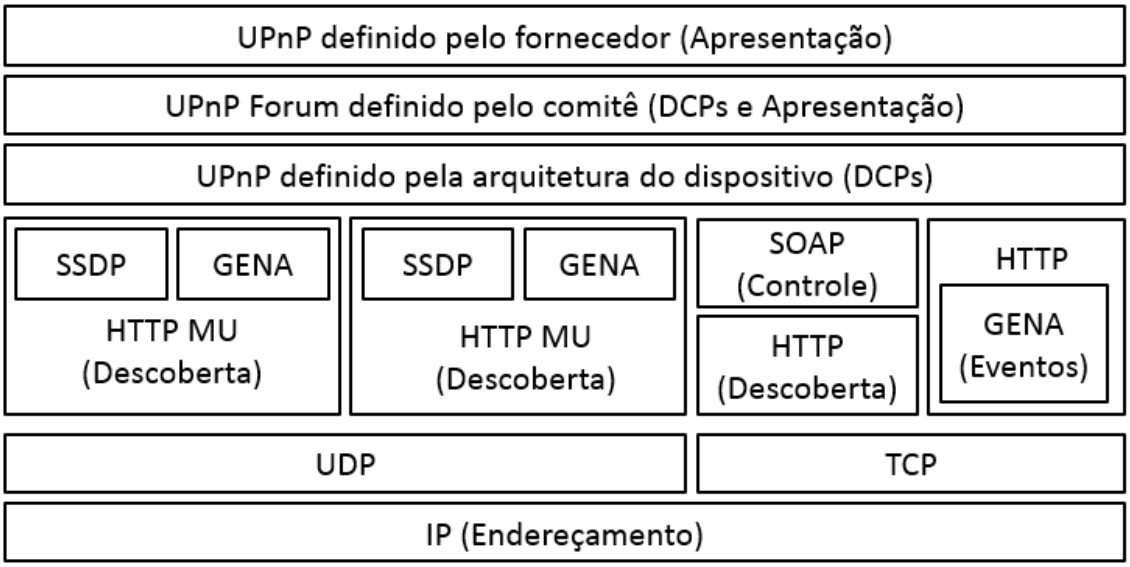
2.3.1.1 Universal Plug and Play

Universal Plug and Play (UPnP) é uma tecnologia de aplicação em camadas que é suportada pela pilha do protocolo TCP/IP nas camadas inferiores de comunicação com o objetivo de permitir que os dispositivos se conectem transparentemente e que equipamentos de diferentes fabricantes sejam integrados (ZHANG; YU; XIE, 2011;

UPNP FORUM, 2000; KIM et al., 2002). Essa é uma tecnologia baseada em *web*, e utiliza diferentes protocolos como *Internet Protocol* (IP), *Transmission Control Protocol* (TCP), *User Datagram Protocol* (UDP), *Hyper Text Transport Protocol* (HTTP), *Simple Object Access Protocol* (SOAP), *Extensible Markup Language* (XML), G*eneral Event Notification Architecture* (GENA), etc. Seu objetivo é promover conectividade transparente sem a necessidade de configuração e com descoberta automática de aparelhos (ZHANG; YU; XIE, 2011; MILLER et al., 2001; UPNP FORUM, 2000; FERREIRA; CANEDO; SOUSA JUNIOR, 2013). O protocolo multicamadas do UPnP é apresentado na Figura 4.

Na arquitetura UPnP existem dois tipos de dispositivos, os dispositivos controlados (*Controlled Devices* - CDs) e pontos de controle (*Control Points* - CPs). CDs são nós de redes que fornecem serviços e CPs são controladores que descobrem e controlam equipamentos. Um dispositivo pode ser um CD, um CP ou ambos ao mesmo tempo. Quando um CD é adicionado à rede, este automaticamente obtém um endereço IP, o chamado Endereçamento, e então transmite seus serviços para os CPs via um *multicast Simple Service Discovery Protocol* (SSDP). Por outro lado, quando um CP é adicionado à rede, este começa a procurar equipamentos de interesse na rede (Descoberta).

Figura 4 - Protocolo multicamada UPnP



Fonte: adaptado de UPnP Forum (2001), UPnP Forum (2014) e Miller et al. (2001)

A interação UPnP é baseada em seis passos: Endereçamento, Descoberta, Descrição, Controle, Eventos e Apresentação (ZHANG; YU; XIE, 2011). O Endereçamento é a obtenção automática de um endereço IP. A Descoberta é realizada por SSDP. A Descrição é realizada por meio da descrição dos serviços de um dispositivo, e utiliza um modelo de dispositivo UPnP e modelos de serviços padrões, como liga/desliga, dimerização, impressa, entre outros. O controle pode ser realizado através mecanismos distintos, entre eles a (i) Invocação de Ação, chamada de procedimento remoto, (ii) Mensagem de Controle, mensagem *Simple Object Access Protocol* (SOAP) no formato *eXtensible Markup Language* (XML) via pedido *Hypertext Transfer Protocol* (HTTP *request*) e (iii) Resposta via HTTP (HTTP *response*). O mecanismo de Eventos é a inscrição do CP para um serviço, para que seja notificado quando as variáveis do serviço mudarem. A Apresentação é uma interface *web* acessível via navegador com uma descrição em linguagem humana dos serviços do equipamento escrita pelo fabricante, na qual o usuário tem o controle direto do equipamento através da *Uniform Resource Locator* (URL).

Essa interação começa uma vez que um CP descobre um CD. O CP obtém a descrição do serviço do CD a partir da URL do dispositivo (Descrição), divulgado no broadcast do CD. Se esses serviços correspondem aos requisitos do CP, o dispositivo pode se subscrever a ele (Evento) ou utilizar o serviço para controlar o dispositivo diretamente (Controle) dependendo da natureza do serviço. O usuário tem a opção de entrar diretamente na interface *web* através de um navegador para acessar a descrição e os controles do dispositivo.

Em um ambiente residencial, o roteador Wi-Fi precisa ter o serviço de UPnP ativado. Os roteadores atuais possuem essa funcionalidade, porém ela vem desabilitada. Quando ativa, possibilita que os CPs utilizem suas funções de descoberta de serviços e que os CDs transmitam seus serviços. Uma vez que o CP descobre o CD, ele acessa diretamente seu serviço, pois não há um controlador central nessa arquitetura. O roteador pode ser utilizado como um repositório de serviços, mapeando-os aos dispositivos provedores.

Com essa definição da arquitetura UPnP, o Fórum UPnP (MILLER et al., 2001) iniciou o desenvolvimento de protocolos padronizados para o controle de dispositivos (*Device Control Protocols* - DCPs), comunicações e métodos de interação padrão para aparelhos eletrônicos. Nos últimos anos, comitês

desenvolveram DCPs para sistemas de áudio e vídeo, automação residencial, segurança, impressoras, câmeras e *gateways* de Internet. Um grupo de 20 membros eleitos é responsável por organizar, aprovar e publicar os DCPs em seu *website*[1].

Entre os protocolos propostos recentemente, tem-se o apresentado em (MITSUGI et al., 2014; MITSUGI; YONEMURA; YOKOISHI, 2013), que resolve o problema de integração entre o UPnP e o *Object Naming Service* (ONS). ONS é um repositório global de serviços mantido pela GS1 baseado na identificação global única EPC e chave GS1. De acordo com o autor, nesse protocolo, um CP pode coletar identificadores dos dispositivos ativos, e os serviços associados a esses dispositivos podem ser obtidos automaticamente pelo ONS. Esse mecanismo facilita o desenvolvimento de aplicações para residências inteligentes pois a lista atualizada de dispositivos disponíveis, mesmo dispositivos restringidos em redes limitadas, está pronta para ser utilizada pelos desenvolvedores a qualquer momento.

Outra aplicação recente do UPnP envolve um método de interconexão entre redes IGPRS e redes UPnP baseado na plataforma do *Open Service Gateway Initiat*ive (OSGI) com o objetivo de integrar os dispositivos de ambos os padrões (TAN et al., 2015).

2.3.1.2 Digital Living Network Alliance (DLNA)

O padrão DLNA pode ser considerado como uma camada intermediária baseada na estrutura original de rede UPnP AV (DLNA, 2007). A DLNA é uma organização comercial organizada pela Sony em junho de 2003 para, colaborativamente, definir orientações para interoperabilidade com o objetivo de permitir o compartilhamento de mídia digital entre aparelhos multimídia. O DLNA tem em vista a conexão da rede cabeada e de rede sem fio de computadores, fiações e equipamentos móveis, a fim de permitir o compartilhamento e aumento dos meios de comunicação e serviços digitais de conteúdo (LI et al., 2008).

Na DLNA *Home Networked Device Interoperability Guidelines* a arquitetura UPnP AV é adotada para o gerenciamento de mídia e controle entre os aparelhos (ZHANG; YU; XIE, 2011). Portanto, o DLNA utiliza o UPnP para gerenciamento, descoberta e controle de mídia nos aparelhos distribuídos. Esse padrão define duas

---

1 www.upnp.org

classes de dispositivo: servidores (DMS) e *players* (DMP) de mídias digitais. O servidor busca adquirir, gravar, armazenar e compartilhar conteúdo de mídia, enquanto o *player* busca na rede residencial os conteúdos de mídia para serem reproduzidos.

A implementação de um dispositivo para compartilhar em uma rede residencial os recursos de mídia na rede baseado em DLNA é chamado de Adaptador de Mídia Digital (*Digital Media Adapter*, DMA) e está apresentado em (LI et al., 2008). Ele pode acessar arquivos de áudio e vídeo em outros computadores, decodificá-los no DMA, e exibi-los em outro dispositivo de exibição. Neste cenário, (YIM; LEE, 2012) propõe um mecanismo de eficiência baseado no DLNA que se destina a esconder a heterogeneidade de hardware, software, formatos de dados e protocolos de comunicação. A arquitetura proposta permite que os aplicativos dos usuários interajam com dados de uma ampla gama de serviços e dispositivos de rede utilizando uma interface de alto nível e abstrata que segue os padrões de serviços *web*. Por outro lado, (WANG et al., 2014) considera que DLNA não leva em consideração a economia de energia e propõe um mecanismo de eficiência energética para DLNA (E2-DLNA), que pode alcançar o equilíbrio entre a qualidade de serviço (QoS) e consumo de energia. A ideia é reduzir a quantidade de mensagens *multicast* desnecessárias.

2.3.1.3 Konnex (KNX)

Konnex é a fusão de três associações Europeias, BCI (BatiBUS Clube Internacional), EIBA (*European Installation Bus Association*) e EHSA (*European Home System Association*), para criar um padrão aberto unificado para automação de residências e prédios na Europa. A associação Konnex foi criada em 1999 e era composta por nove empresas da área de tecnologia, como Bosh Telecom, Electrolux e Siemens. No início dos anos 2000, ela era composta por mais de 200 empresas do mundo, fabricantes de produtos eletrônicos e provedores de serviços de telecomunicação. A primeira especificação KNX foi publicada em 2002 baseada no padrão EIB (*European Installation Bus*) da EIBA (RUTA; SCIOSCIA; DI SCIASCIO; LOSETO, 2011a).

Konnex, ou simplesmente KNX, é um padrão aberto importante para o controle de edifícios e residências. Ele abrange toda a gama de opções de comunicação residencial, incluindo par trançado (TP), linha de energia (PL), radiofrequência (RF) e Ethernet (IP) (KYSELYTSYA; WEINZIERL, 2006). KNX une e reforça três padrões de barramento legado, EIB, EHS (European Home System), e Batibus (RUTA, 2014).

Um meio de transmissão no KNX (TP, RF, PL ou IP) pode interligar canais heterogêneos de dispositivos tornando-os capazes de compartilhar e trocar informações de forma padronizada. Cada componente KNX pode ser visto como um sensor de detecção de uma situação do ambiente, tais como estações meteorológicas, sensores de iluminação, sistemas de monitoramento, sensores de presença, ou um atuador que controla equipamentos, como cortinas ou persianas, aparelhos de segurança, gestores de energia, aquecimento, dispositivos de ventilação, sistemas de ar-condicionado e multimídia. Cada aparelho pode ser monitorado, supervisionado e sinalizado através de um protocolo compartilhado sem a necessidade de um centro de controle adicional (KNX, 2006).

Ao contrário do UPnP que proporciona características de *plug and play*, a sub-rede Konnex deve ser instalada e configurada com ferramentas de software antes de estar apta a funcionar corretamente (MIORI et al., 2006). Ela utiliza XML para armazenar e trocar configurações e informações na rede KNX.

O padrão KNX é baseado no conceito de *datapoints*. Entradas, saídas e parâmetros são instâncias de *datapoints* de objeto, grupos ou propriedades. Esses *datapoints* são padronizados e agrupados em blocos funcionais e recebem um nome e um tipo de um formato de dados. O KNX possui quatro modos de operação: (i) modo de Sistema, (ii) modo de Configuração, (iii) modo *Push-button* e (iv) modo *Logical tag*, para diferentes cenários de uso.

O KNX possui uma topologia livre e descentralizada, com inteligência distribuída, que pode detectar, controlar e monitorar as funções e sequências através de uma linha de comunicação unificada. A arquitetura Konnex é escalável (BOLZANI, 2006), permitindo o crescimento do sistema para se adaptar às necessidades das residências e prédios, e pode fornecer uma comunicação de ponta a ponta, deixando que os sensores e atuadores se comuniquem com baixo tempo de resposta em situações de controle dinâmico.

O *Advanced Network technology for Unified Building Integration* (ANubis) é uma iniciativa destinada ao desenvolvimento de padrões abertos, para fornecer integração do KNX com os padrões utilizados na Internet. Seu objetivo é a implementação econômica, desde pequenas residências até grandes edifícios, utilizando barramentos hierárquicos. Cada barramento pode ter 12 linhas, e cada linha, 255 dispositivos. Um acoplador pode conectar até 15 barramentos (BOLZANI, 2006).

Todas essas características fazem o Konnex adequado para a automação residencial (MIORI et al., 2006; MIORI, RUSSO e ALIBERTI, 2010), e sua linha comercial é o Instabus. Apoiado pela Siemens e certificado pelo EIBA, ele é focado no consumo de energia residencial e fornece soluções de controle da luz, controle de sistemas de aquecimento, ventilação e condicionamento de ar, e equipamentos de segurança. É um subsistema distribuído que possui um servidor central que conecta a sub-rede KNX com a rede na qual o cliente KNX está conectado (RUTA, 2014). O usuário pode configurar, programar e administrar os aparelhos do sistema com um computador conectado utilizando uma interface serial RS232 e o software EIB.

Nos últimos cinco anos os pesquisadores Ruta; Di Sciascio; Scioscia, e Loseto (2011a) divulgaram artigos relacionados ao KNX, propondo: (i) a evolução semântica baseada no protocolo KNX capaz de interagir usuários (cujas necessidades são expressas através de seus perfis gravados) com uma infraestrutura de eletrodomésticos, baseada em ontológica, de uma forma automatizada, (ii) introdução de uma extensão, baseada na semântica, da camada de aplicação da norma KNX, preservando aplicativos legados e abrindo possibilidades para outras melhorias (RUTA; SCIOSCIA; DI SCIASCIO; LOSETO, 2011b), e (iii) uma abordagem baseada na semântica capaz de criar uma interface entre usuários e dispositivos, cujas características são expressas por meio de perfis gravados, dentro de uma infraestrutura avançada de residência (RUTA, 2012).

2.3.1.4 LonWorks©

LonWorks© significa *Local Operating Networks*, e é uma plataforma criada pela empresa Echelon Corporation em 1991, baseada no protocolo LonTalk©, para

aplicações de controle (ECHELON, 2014). Todo dispositivo LonWorks© inclui um Neuron Chip, um *transceiver* e os circuitos eletrônicos requeridos a uma dada aplicação (YAN; BO, 2013). O Neuron Chip é um circuito integrado com vários microprocessadores, memória RAM, ROM e portas de interface de entrada e saída que implementa o protocolo LonTalk©. O *transceiver* é um módulo eletrônico que fornece a interface física para se comunicar com outros dispositivos. O conceito associado a essa tecnologia é a constituição de sistemas de automação baseados em dispositivos ou em grupos de dispositivos (denominados nós) com capacidade de processamento local, que podem ser interligadas por intermédio de diversos tipos de meios de comunicação (par trançado, radiofrequência, fibra óptica, etc.) criando, assim, uma rede (KORENCIAK, 2012).

O protocolo LonTalk© inicialmente era de propriedade da Echelon Corporation, porém veio a se tornar um padrão aberto, para poder ser utilizado por outras empresas para a fabricação de equipamentos seguindo esse padrão. Ele foi padronizado em 1999 recebendo a denominação ANSI/CEA-709.1-B, e depois em 2008 com a ISO/IEC 14908 (BOLZANI, 2006).

O protocolo LonTalk© segue o modelo de referência *Open Systems Interconnect* (OSI) e fornece serviços em nas sete camadas do modelo. Ele é um protocolo complexo e completo, e cada camada possui suas funcionalidades definidas (BOLZANI, 2006).

A primeira camada é a Física, que permite utilizar múltiplos meios físicos de comunicação, por exemplo, RS-485, *Free Topology* (FT), ou rede elétrica (*Power Line Communication* - PLC), tecnologia com ou sem fio e fibra óptica.

A segunda camada é a Data Link, e utiliza o algoritmo diferente do *Media Access Control* (MAC) para fornecer o controle independentemente dele, baseado em melhorias do *Carrier Sense Multiple Access* (CSMA), conceitos de *p-persistent* CSMA e *non-persistent* CSMA. Além disso, pode utilizar *p-persistent* CSMA preditivo para fazer a priorização de acesso. Além do algoritmo MAC, essa camada oferece a verificação de redundância cíclica (*Cyclic Redundancy Check* - CRC) utilizando o padrão CCITT CRC-16 para codificação dos bits.

A terceira camada é a de Rede que utiliza endereçamento hierárquico, e os endereços seguem a seguinte máscara: primeiramente o domínio do nó, em seguida a sub-rede e por último o número de identidade do nó. Essa máscara foi criada para

oferecer um mecanismo simples de endereçamento para permitir um roteamento de pacotes e manter uma pequena tabela de roteamento.

A quarta camada é a de Transporte, que prevê a retransmissão de pacotes e detecção de duplicidade de pacotes. O protocolo utilizado permite uma transmissão confiável, tanto ponto a ponto como ponto a multiponto. Assim, a comunicação é transparente para os dispositivos das aplicações independentemente da mensagem a ser enviada para um ou mais recebedores. Ademais, a camada de transporte tem o controle do tempo de resposta para atender as exigências de aplicação.

A quinta camada é de Sessão. Os equipamentos e aplicações a utilizam para realizar o pedido e resposta do serviço. O protocolo salva a mensagem em caso de perda da mensagem, e pode também recuperá-la e enviá-la novamente sem a necessidade de intervenção da aplicação.

A sexta camada é a de Apresentação, na qual o padrão *publisher-subscriber* da Echelon é utilizado. O cabeçalho da mensagem codifica a semântica dos dados passados para a camada de aplicação seguindo os padrões da Echelon.

A sétima camada é a de Aplicação, na qual a lógica dos dispositivos é construída e corresponde ao centro dos dispositivos. Essa camada é responsável pelo controle interno do dispositivo, processamento das entradas, tomadas de ação e gerenciamento de periféricos.

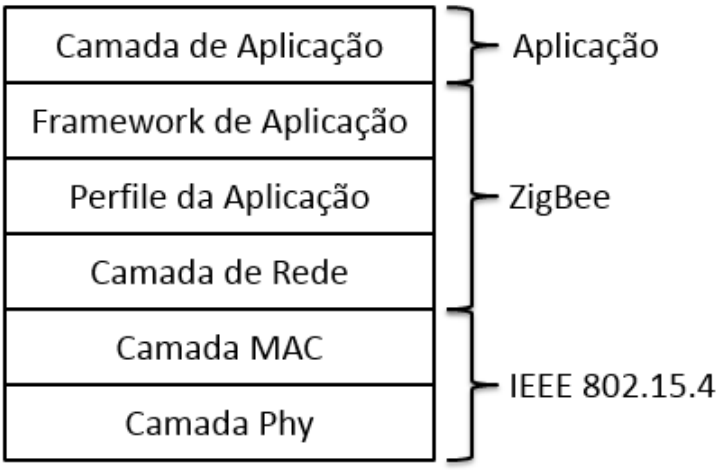
No entanto, de acordo com Fisher (2000), fornecedores que precisam adicionar funcionalidade aos seus sistemas não podem fazê-lo sem violar o acordo assinado para participar da associação LonMark©, portanto, perdendo o direito de portar a marca LonMark©, ou convencendo membros Gold a endossar suas extensões. Um problema secundário é que nem todos os sistemas são sistemas LonMark©. É possível e provável que LonWorks continuará a ser usado por alguns fornecedores que não desejam, ou não conseguem pagar, para ser parte do consórcio LonMark©.

2.3.1.5 ZigBee

O padrão ZigBee consistem em quatro camadas: a física (PHY), a *Medium Access Control* (MAC), a de rede (NWK) e a camada de aplicação (APL), como pode ser visto na Figura 5. As camadas PHY e MAC estão definidas no padrão IEEE 802.15.4

e as camadas NWK e APL foram definidas pela Aliança ZigBee. A camada de aplicação contém o perfil público da aplicação, o que permite que os fornecedores OEM criem produtos interoperáveis, como produtos de automação residencial (KALAIVANI; ALLIRANI; PRIYA, 2011; FERREIRA; CANEDO; SOUSA JUNIOR, 2013).

Figura 5 - Protocolo em camadas do ZigBee



Fonte: adaptado de Kalaivani; Allirani e Priya (2011)

Um grupo de empresas lideradas pela Philips começou a desenvolver uma tecnologia *Wireless Personal Area Network* (WPAN) devido ao atraso na definição do padrão Bluetooth. A nova tecnologia, chamada ZigBee, visava um baixo custo e um baixo consumo de energia, e é também conhecido como PURLNet, HomeRF Lite, Firey, RFEasyLink ou RF Lite (BOLZANI, 2006).

A Aliança ZigBee, composta pela Mitsubishi, Honeywell, Invensys, Motorola e Philips, é responsável por definir e desenvolver aplicativos, programas de certificação, logotipos, estratégias de marketing e protocolos além da camada MAC. A fundação do ZigBee foi o padrão IEEE802.15.4, que especifica as camadas físicas que podem ser usadas e como usá-las. O padrão físico para radiofrequência opera em bandas não licenciadas em todo o mundo, a 2.4GHz (global), 915Mhz (Américas) e 868 MHz (Europa) (ZIGBEE ALLIANCE, 2014). Ele pode entregar dados em taxas de transferência de 250 kbit/s em 2.4GHz (16 canais), 40 kbit/s em 915 MHz (10 canais) e 20 kbit/s em 868MHz (1 canal).

Os primeiros módulos chegaram ao mercado em 2003 a um baixo custo, cerca de 2 dólares (BOLZANI, 2006), para ser incorporado a um eletrodoméstico de

forma a torná-lo um equipamento de informação e interconectá-lo a outros equipamentos. O módulo ZigBee pode operar em três diferentes modos, *Master-Slave*, *Peer-to-Peer* e *Mesh*, e os equipamentos ZigBee podem ser de três tipos, Coordenador ZigBee (*ZigBee Coordinator* - ZC), Roteador ZigBee (*ZigBee Router* - ZR) e Dispositivo Final ZigBee (*ZigBee End Device* - ZED).

De acordo com a Aliança ZigBee, a padronização da interoperabilidade do equipamento permite que fabricantes de produtos foquem na inovação de seus produtos em vez de tentar criar seus próprios protocolos de rede exclusivos (ZIGBEE ALLIANCE, 2014), e isso ocorre também com outras tecnologias. Se o mercado utilizar apenas um padrão, flexível o bastante para suportar todos os requisitos, todos os equipamentos podem ser facilmente integrados e o problema de interoperabilidade poderia ser resolvido. Porém, ter apenas um padrão, forte o suficiente para atender todas as necessidades e requisitos de todos os equipamentos seria difícil, se não impossível, e a parte mais difícil seria persuadir os fabricantes a utilizar um padrão unificado.

2.3.1.6 X-10

O X-10 é um protocolo utilizado para controlar remotamente transmissores e receptores por meio da fiação elétrica já existente na residência (BOLZANI, 2006; VALTCHEV; FRANKOV, 2002). O sistema foi introduzido pela Sears Home Control System e pela Radio Shark e Plug and Play System. Posteriormente, outras empresas, como a General Electric e Philips também o adotaram. Para uniformizar o sistema X-10, alguns módulos foram patenteados e suas licenças foram oferecidas por um preço razoável (BOLZANI, 2006).

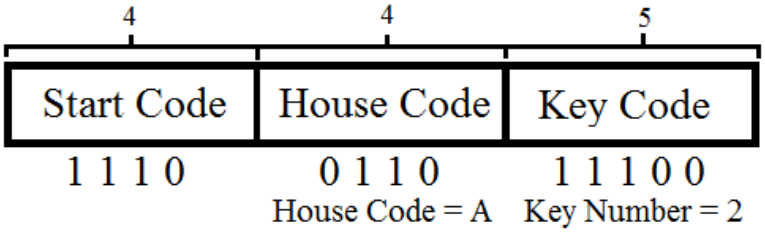
Transmissores transmitem comandos precedidos pela identificação da unidade receptora através do ponto de acesso da linha de energia, como uma tomada, um soquete ou um conector de tomada. Os receptores começam a ler a mensagem através da unidade de identificação, e se este for seu próprio identificador, o receptor continua lendo a mensagem inteira para processar o comando.

O padrão X-10 define 256 endereços, 16 códigos de unidade para cada um dos 16 setores (A-P), o endereço total tem a seguinte máscara 255.15.15. Se a

comunicação em uma área maior que 185 m^2 apresentar problemas de transmissão será necessário criar uma ponte, um ampliador ou um repetidor de sinal.

Esse padrão define como os bits de uma mensagem são transmitidos pelas linhas de energia utilizando sinal de corrente alternada como uma portadora de sinal. O cabeçalho de uma mensagem começa com um *Start Code* de quatro dígitos, 1110, depois quatro dígitos para *House Code*, para especificar o setor (A-P), e, então, os últimos cinco dígitos para *Key Code*, que especifica a unidade receptora da mensagem. A mensagem de comando vem depois do cabeçalho. A Figura 6 apresenta o cabeçalho do protocolo X-10.

Figura 6 - Cabeçalho do protocolo X-10



Fonte: adaptado de Bolzani (2006) e Valtchev e Frankov (2002)

Porém, este meio de transmissão impõe na rede limitações de velocidade e, portanto, diminui a taxa de transferência de dados. Assim, o X-10 é utilizado para aplicações como controle de iluminação, eletrodomésticos conectados e sensores de segurança (VALTCHEV; FRANKOV, 2002).

Outra abordagem que avançou a utilização da rede elétrica para comunicação foi o *Commom Application Language* (CAL). Este foi desenvolvido pelo CEBus como uma linguagem de comunicação universal para eletrodomésticos e eletroeletrônicos que podem ser interpretados por todos os equipamentos de uma residência, permitindo a comunicação através das linhas de energia. Essa linguagem foi adotada pelo padrão EIA 721 e especificado pelo *Home Plug and Play* como um guia de referência (BOLZANI, 2006).

O Home Plug and Play expandiu o CAL e o tornou um conjunto de regras e ferramentas que permitem a comunicação entre diferentes dispositivos através de diferentes interfaces para diferentes camadas físicas e protocolos. Em 1995, um

grupo de empresas formado pela Honeywell, Intel, Microsoft, Thomsom Consumer Eletronics, entre outras, criou a Força Tarefa *Home Plug and Play* para desenvolvê-lo e promovê-lo para uma maior adoção no mercado. O objetivo era oferecer ao mercado um ambiente integrado com uma solução para interconexão e comunicação entre dispositivos de áudio e vídeo, luz, climatização, controle de consumo de energia, segurança e rede. A Figura 7 apresenta o protocolo multicamada do *Home Plug and Play*.

Figura 7 - Protocolo multicamadas do Home Plug and Play

Fonte: adaptado de Bolzani (2006)

2.3.2 Arquiteturas

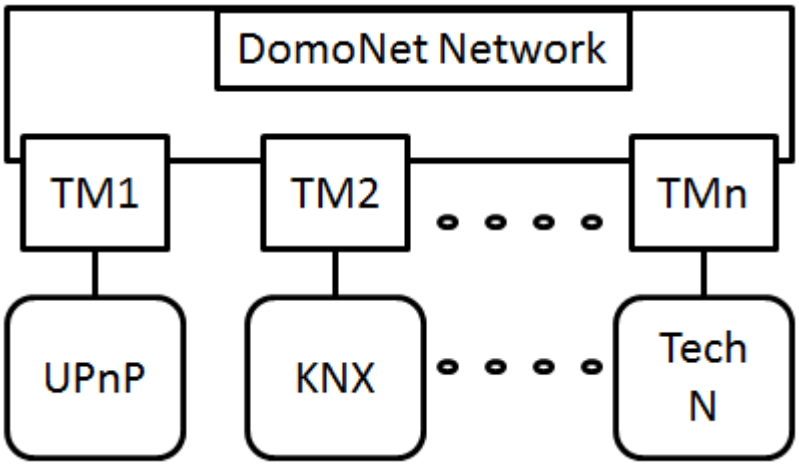
2.3.2.1 DomoNet

A DomoNet é um projeto de pesquisa italiano do Domotics Lab do Instituto de Ciência da Informação e Tecnologias "A. Faedo" (ISTI) do Conselho Nacional Italiano de Pesquisa em Pisa (MIORI, RUSSO e ALIBERTI, 2010). Eles abordaram a questão da integração de uma maneira abrangente e abstrata, ao invés de abordar a tecnologia diretamente usando mapeamentos específicos dos padrões de forma ad hoc. DomoNet é focado na funcionalidade e na semântica, utiliza XML para

descrição, e serviços *web* para controle. Ambas as tecnologias foram escolhidas pela sua natureza multiplataforma e papel emergente como um padrão aberto para organizar e utilizar recursos de aplicação distribuídos (MIORI et al., 2006). A vantagem de utilizar uma solução padrão do W3C é a garantia de compatibilidade entre a aplicação e outros softwares baseado nesse padrão, garantindo assim a liberdade de escolha do sistema de software, linguagem de programação ou da arquitetura do hardware (MIORI; RUSSO, 2014).

A arquitetura DomoNet é baseada no modelo de Arquitetura Orientada a Serviço (SOA), na qual os serviços representam as funções dos dispositivos. Ela consiste em um *gateway* central para interconectar diferentes tecnologias. Cada tecnologia possui a *TechManager* (TM) que fornece serviços *web* para acessar e controlar a funcionalidade de tecnologia específica para a topologia do dispositivo (MIORI et al., 2006). A Figura 8 apresenta o conceito *TechManager*.

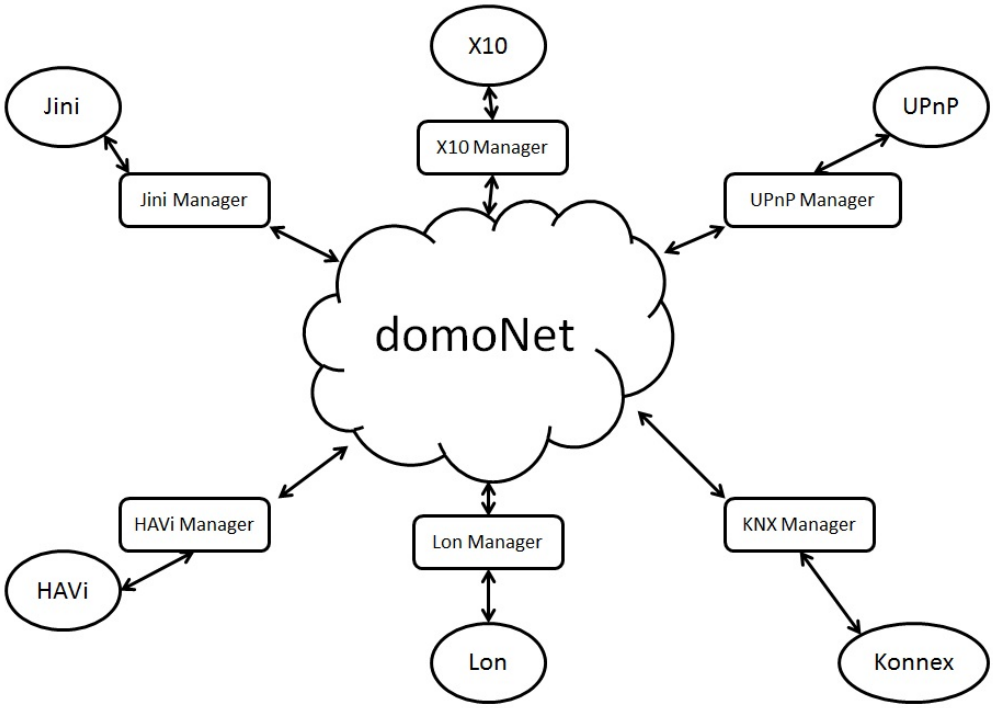
Figura 8 - DomoNet TechManager



Fonte: adaptado de Miori et al. (2006)

O *gateway* DomoNet é responsável por interconectar todos as diferentes infraestruturas de rede necessárias para os equipamentos se comunicarem. Ela define uma camada de controle que pode ser vista como uma meta-infraestrutura de serviços de alto nível criada com um mapeamento modular entre a camada de abstração, os protocolos e as peculiaridades da tecnologia. A Figura 9 apresenta a Arquitetura DomoNet na qual cada sistema domótico representa uma sub-rede que pode ser conectada pelo *TechManager* específico hospedado no *gateway*.

Figura 9 - Arquitetura DomoNet



Fonte: adaptado de Miori et al. (2006)

A ideia de criar uma camada abstrata para integrar os sistemas domóticos que utilize uma tecnologia particular tratando-a como uma sub-rede, criou um conceito importante para o ambiente de automação residencial. Esse conceito permite a melhoria da arquitetura, protocolos, rede e tecnologia com um subsistema domótico para melhor atender as necessidades dos dispositivos. Considerando que um subsistema possui dispositivos similares e que eles possuem requisitos similares, a infraestrutura do subsistema pode ser desenvolvida ou escolhida para melhor atender as necessidades e requisitos dos dispositivos, portanto, especializando a infraestrutura do subsistema para sua funcionalidade.

2.3.2.2 Jini

A palavra "jini" vem da palavra magia em árabe. O objetivo desse padrão é conectar diversos aparelhos diferentes com autoconfiguração e auto instalação, e assim dividindo seus recursos mais facilmente. Jini foi um projeto da SUN Microsystems, que foi comprada pela Oracle em 2010 (ORACLE, 2010), baseado no ambiente

JAVA. Ele é uma coleção de APIs Java e protocolos de rede para construir um sistema de distribuição organizado, chamado federação, que não possui um nó central. Uma falha em um único nó consequentemente não impacta os outros nós e equipamentos do sistema. Essas características fazem com que sistema inteiro seja mais flexível e adaptável.

O Jini herda os benefícios e o poder do Java. As aplicações do Jini são móveis já que utiliza *bytecode* para rodar em uma *Java Virtal Machine* (JVM), e utiliza um paradigma orientado por objetos que permite o reuso de classes e objetos. Do ponto de vista de integração dos usuários, é similar ao UPnP, mas o Jini interage com a parte baixa do protocolo de rede multicamada. A tecnologia do Jini é uma coleção de camadas de software que fornece serviços para dispositivos e computadores baseados no Java *Remote Method Invocation* (RMI) (BOLZANI, 2006) e foi projetado para proporcionar funcionalidades adicionais para os pacotes do Java RMI, assim como um conjunto de APIs, protocolos e convenções (AL-MUHTADI et al., 2000) para facilitar a construção sem esforço de aplicações distribuídas (JAI; OGG; RICCIARDI, 2000) e construir um ambiente doméstico inteligente de modo simples e intuitivo (AL-MUHTADI et al., 2000).

O cliente Jini pode ser embarcado em todos os equipamentos capazes de rodar a JVM e integrá-lo na rede do Jini. Outros dispositivos mais simples também são capazes de se conectar, porém com funcionalidades limitadas (AL-MUHTADI et al., 2000). A arquitetura Jini é baseada em serviços, e os serviços se comunicam através de um conjunto de interfaces Java implementadas:

A descoberta tem uma interface que é utilizada quando um novo equipamento é conectado à rede e tenta tornar seus serviços disponíveis. Nesse procedimento o equipamento transmite sua presença à rede para alcançar os Serviços *Lookup*; esses serviços funcionam como um repositório de indicações de serviços.

A interface de união é utilizada quando os serviços *Lookup* recebem um sinal de um novo serviço e mandam de volta o objeto do serviço de registro, o qual será utilizado por um novo serviço para realizar a união. Depois da união, uma cópia da interface do serviço fica disponível para toda a rede Jini.

O serviço *Lookup* armazena uma lista de interfaces de serviços disponíveis. A interface de *Lookup* é utilizada quando um cliente Jini requer um serviço específico,

o serviço *Lookup* é responsável por encontrar o proprietário do dispositivo do serviço e enviar suas informações ao cliente.

Toda a comunicação em uma rede Jini ocorre utilizando o Java RMI, tornando-se assim uma extensão abstrata do Java RMI. Clientes do Jini necessitam de menos informações sobre a infraestrutura para utilizar os serviços na descoberta do mecanismo quando necessário. O acesso ao serviço ocorre pela locação de uma fatia de tempo e da permissão de acesso a um recurso por um período de tempo (JAI; OGG; RICCIARDI, 2000).

Jini pode ser utilizado em processos Java distribuídos, porém também integra múltiplos sistemas de equipamento como em uma infraestrutura de comunicação e integração. Por exemplo, a empresa ProSyst Software fornece produtos para controle de eletrodomésticos baseado na plataforma Jini (MIORI et al., 2006; MIORI; RUSSO; ALIBERTI, 2010). A plataforma Jini têm as vantagens de sistemas distribuídos, melhora a escalabilidade e garante que a conectividade distribuída seja segura e disponível de forma uniforme. Mas também precisa lidar com as questões de agregação, tais como o controle de grupos de dispositivos, que surgem para alcançar a coordenação entre os dispositivos autônomos (JAI; OGG; RICCIARDI, 2000).

2.3.3 Iniciativas

2.3.3.1 Projeto HYDRA

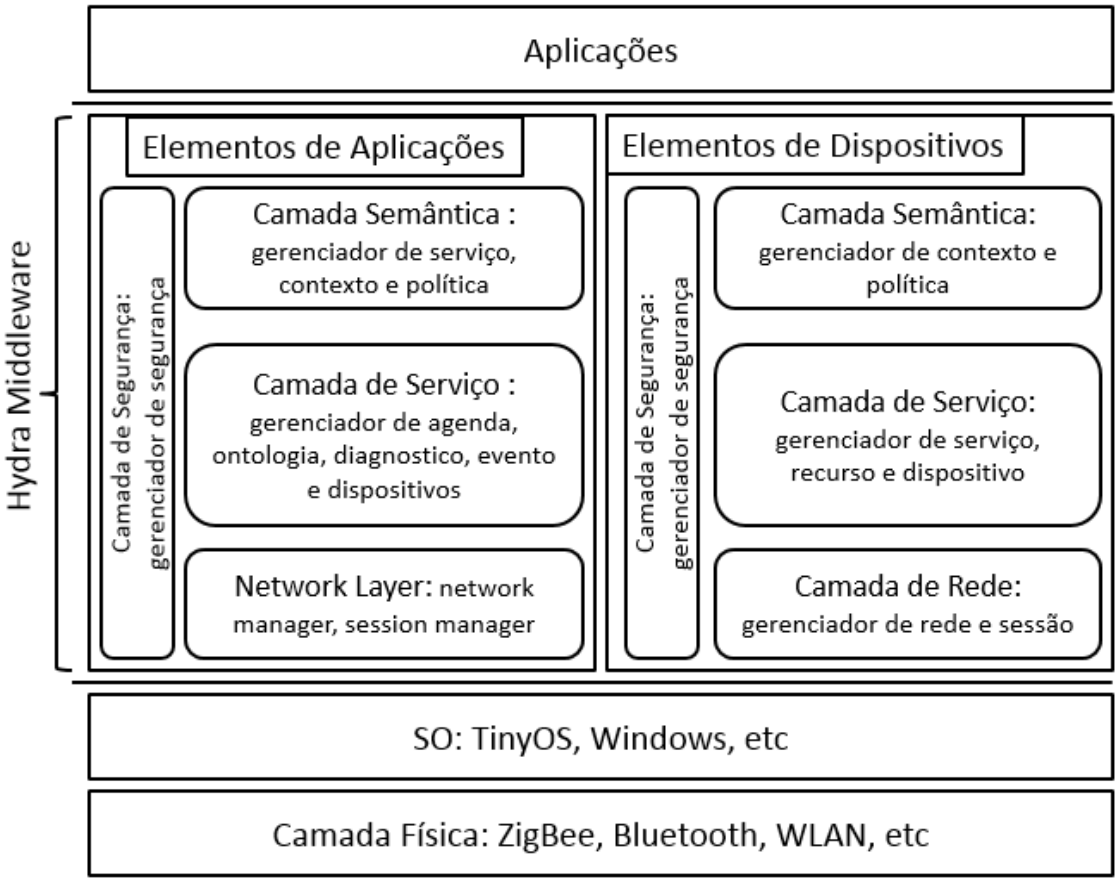
HYDRA é um *middleware* para sistemas inteligentes embarcados conectado a uma rede de arquitetura orientada a serviço. Ele pode ser implantado em uma nova rede ou em uma já existente utilizando equipamentos distribuídos com e sem fio que podem ter recursos limitados, como consumo de processamento, memória e energia.

O Projeto Hydra, *Networked Embedded System Middleware for Heterogeneous Physical Devices in a Distributed Architecture*, foi definido no IST-2005-034891 e financiado pelo programa *Framework* FP6 da *Information Society Technologies* (IST) entre 2002 e 2006 (HYDRA, 2014). A visão deste projeto foi criar o *middleware* que possa ser implantado amplamente para sistemas embarcados inteligentes em rede para permitir o desenvolvimento de aplicações embarcadas

rentáveis e inovadoras para novos equipamentos e aqueles já existentes (SARNOVSKÝ et al., 2008). A visão de futuro descrita no projeto possui um ambiente com diferentes equipamentos inteligentes interligados que estarão em todos os lugares que se for, criando uma rede inteligente ubíqua.

O desenvolvimento é dividido em três domínios, automação de construções, assistência médica e agricultura. A estrutura funcional, presente em todos os domínios, é dividida em duas camadas, camada de elementos de aplicação e camada de elementos do dispositivo. Elementos de aplicação possuem mais poder de processamento enquanto os elementos de dispositivo possuem um módulo HYDRA incorporado com funcionalidade limitada. A Figura 10 apresenta a estrutura do *middleware* HYDRA com os elementos de aplicações e elementos de dispositivos.

Figura 10 - Protocolo multicamadas do HYDRA



Fonte: adaptado de Sarnovský et al. (2008)

Um kit de desenvolvimento de software (SDK) foi disponibilizado para criar aplicações e dispositivos para esse sistema. Nesse sistema, cada dispositivo possui

um cliente HYDRA incorporado com o papel de *proxy* entre o dispositivo e o *middleware* HYDRA, enquanto o *middleware* possui o papel de um intermediário, o *broker*. Para essa integração funcionar, o *middleware* e os dispositivos precisam estar conectados à rede residencial.

A arquitetura HYDRA por ser baseada no conceito de *middleware*, no qual existe um nó central (*middleware)* que orquestra os nós circundantes (equipamentos), torna-se dependente desse nó central. Além disso, esse *middleware* precisa ser capaz de suportar todos os diferentes dispositivos da residência, e de sustentar toda a troca de dados entre todos os dispositivos conectados.

2.3.3.2 Amigo

Amigo tem como objetivo proporcionar recursos de ambiente inteligente, focado em ambientes residenciais em rede e na usabilidade do sistema. Suas diretrizes incluem interfaces amigáveis para o usuário, interoperabilidade, e descoberta automática de equipamentos e serviços (MIORI; RUSSO; ALIBERTI, 2010). Esta interoperabilidade entre diferentes domínios de aplicação também pode ser estendida em diferentes residências e locais.

Amigo é um projeto integrado europeu composto por quinze empresas e institutos de pesquisa líderes em informação e comunicação na Europa, nas áreas de mobile, redes residenciais, desenvolvimento de software, eletrônicos e equipamentos residenciais. Alguns dos participantes são a Philips, a Telefônica, a France Telecom e a Fraunhofer. O projeto foi definido no IST-2002 -2.3.2.3 (*Open Development Platforms for Software and Services*) e foi financiado pelo programa FP6 (FP6-2003-IST-2) da *Information Society Technologies* (IST) entre 2004 e 2008 e recebeu cerca de 24 milhões de euros (EUROPEAN COMMISSION, 2014; JANSE; VINK; GEORGANTAS, 2008).

De acordo com Janse; Vink e Georgantas (2008), o projeto Amigo suporta um grande número protocolos existentes na área de Orientação a Serviços para a descoberta e para a comunicação de uma forma interoperável. Isso torna possível para os programadores selecionar o protocolo de sua escolha mantendo as

funcionalidades dos serviços, mesmo para aqueles que utilizam protocolos diferentes.

Uma aplicação comum do Amigo é, por exemplo, um recurso de rastreamento dos membros de uma família, com o objetivo de alertar a pessoa apropriada quando alguém ficar doente, um recurso para planejar uma refeição em família e encomendar os ingredientes do supermercado ou um recurso para alertar o habitante de residência quando ele esquecer sua carteira, entre outros, para tornar o ambiente residencial mais inteligente.

2.3.3.3 Grupo Open Building Information Exchange (oBIX)

O Grupo oBIX é composto por líderes da indústrias e associações que trabalham juntas para criar um padrão em XML e guias de serviços *web* para facilitar a troca de informações entre prédios inteligentes (NEUGSCHWANDTNER, M.; NEUGSCHWANDTNER, G.; KASTNER, 2007; OBIX, 2014), assim como a troca de informação dentre os sistemas mecânicos e elétricos em uma automação predial (MIORI, RUSSO e ALIBERTI, 2010). Esse esforço tem como o objetivo permitir a integração de aplicações empresariais e trazer uma verdadeira integração de sistemas.

O principal requisito do oBIX é desenvolver uma sintaxe XML comum para representar a informação de diversos sistemas M2M. A filosofia de design do oBIX é baseado em um modelo de dados pequeno, mas extensível, que mapeia todas as informações dos sistemas para uma sintaxe XML fixa simples (CONSIDINE; BOGEN, 2014).

O oBIX inclui profissionais de segurança, HVAC, automação predial, protocolo aberto e disciplinas de TI, foi criado em abril de 2003 como o CABA XML/ *Web Services Guideline Comitte* e hoje é um comitê técnico no *Organization for the Advancement of Structured Information Standards* (OASIS).

2.3.4 Resumo das tecnologias

Neste subcapítulo, tecnologias, padrões, arquiteturas e iniciativas foram apresentadas, analisadas e suas características apontadas. Mas para melhor

compreender o cenário da automação residencial, precisa-se entender como se encaixam e se comportam juntos a partir de uma comparação entre eles.

Essas tecnologias estão disputando o mercado, mas a maneira como os usuários irão escolher qual tecnologia é mais apropriada para suas residências depende de quanto essas tecnologias estão sendo divulgadas e como seu marketing está sendo feito.

2.3.4.1 Comparação entre tecnologias

Este subcapítulo apresenta uma tabela de comparação qualitativa entre as tecnologias e padrões analisados nas seções anteriores, apresentada no Quadro 2. A tabela de comparação foi baseada nas tabelas comparativas apresentadas em (CHEN et al., 2014; WACKS, 2012; TSAI et al., 2014; RAYCHOUDHURY et al., 2013). Os critérios de análise das tabelas de comparação desses artigos foram unidos e aplicados nas tecnologias aqui revisadas, seguindo as referências específicas usadas nas seções anteriores.

A tabela é composta por onze critérios. O primeiro critério é o tipo de comunicação que cada padrão utiliza para automação residencial e o segundo indica se a arquitetura utilizada é capaz de usar a infraestrutura existente na residência ou se é necessária uma nova para interconectar os dispositivos. O critério de tipo de dados mostra as limitações de dados e os tipos de dados para os quais o padrão foi criado.

Os próximos critérios foram classificados em três grupos: Baixa, Média e Alto. As tecnologias foram classificadas nessas métricas de comparação de forma a limitar a escala pelas tecnologias com o maior nível e menor nível no critério analisado. E assim, colocando as outras tecnologias nos grupos de acordo com seu nível naquele critério.

O critério de custo é uma comparação entre os custos das tecnologias voltadas para a compra de um sistema de automação residencial, que vão desde o padrão mais barato até o mais caro analisado. Critérios como complexidade, capacidade de processamento, taxa de transmissão de dados e consumo de energia são relacionados com as limitações das tecnologias e também são categorizadas em três grupo, baixo, médio e alto.

O critério nível de abstração do padrão é relacionado com o quanto a comunicação do padrão é abstrata ou especializada, isto é, uma comunicação de alto ou baixo nível.

Os critérios, no Quadro 2, se encontram na primeira linha, enquanto os nomes das tecnologias se encontram na primeira coluna. Os demais campos são preenchidos com os grupos da categorização realizada e suas siglas seguem a legenda na parte inferior do quadro.

Quadro 2 - Comparação qualitativa

Nome do Padrão	Tipo de Comunicação	Infrastrutura	Tipo de dados	Custo	Complexidade	Poder de Processamento	Taxa de transferência de dados	Consumo de energia	Segurança	Privacidade	Nível de Abstração	Características Desejadas
UPnP	**SOA**	**Existente**	**Qualquer**	**Baixo**	**Baixo**	Médio	**Alto**	Médio	**Alto**	Médio	**Alto**	7
DLNA	**SOA**	**Existente**	AV	**Baixo**	**Baixo**	Médio	**Alto**	Médio	Médio	Baixo	**Alto**	5
LonWorks	OSI	Novo	**Qualquer**	Alto	Alto	**Alto**	**Alto**	Alto	**Alto**	**Alto**	Médio	5
ZigBee	Mesh	Novo	**Qualquer**	Médio	Médio	Médio	Médio	**Baixo**	**Alto**	**Alto**	Médio	4
X10	Rede elétrica	**Existente**	Sensor e Controle	**Baixo**	**Baixo**	Baixo	Baixo	**Baixo**	Baixo	Baixo	Baixo	4
domoNet	Centralizado	Novo	**Qualquer**	Médio	Médio	Médio	**Alto**	Alto	Médio	Baixo	**Alto**	3
Amigo	Middleware	Novo	**Qualquer**	Médio	Médio	Médio	**Alto**	Médio	**Alto**	Médio	Médio	3
Project HYDRA	Centralized	Novo	Sensor e Controle	Médio	Médio	Médio	**Alto**	Médio	Médio	Médio	**Alto**	2
Home Plug and Play	Rede elétrica CAL	Novo	**Qualquer**	Médio	Médio	Médio	Médio	Médio	Baixo	Baixo	**Alto**	2
oBIX	**SOA**	Novo	**Qualquer**	Médio	Médio	Médio	Médio	Médio	Médio	Médio	Médio	2
Konnex	Datapoints	Novo	**Qualquer**	Médio	Médio	Médio	Médio	Alto	Médio	Médio	Médio	1
Jini	Distribuido	Novo	**Qualquer**	Médio	Médio	Médio	Médio	Médio	Médio	Médio	Médio	1
Desejável para Automação Residencial	SOA	Existente	Qualquer	Baixo	Baixo	Alto	Alto	Baixo	Alto	Alto	Alto	

Fonte: autor

Legenda: AV – áudio e vídeo, SC – Sensor e Controle.

A comparação destaca que essas tecnologias têm características diferentes. Dependendo do tipo, complexidade e tamanho do dispositivo ou sistema, uma tecnologia específica pode tornar-se mais adequado para o caso de uso estudado. No entanto, o foco em automação residencial cria limites e limitações dessas

tecnologias, apesar de uma ampla gama de tecnologias de ainda ser capaz de atender a essas características restritas.

O primeiro e segundo critérios, arquitetura e infra-estrutura, mostra duas características dos padrões, mas eles estão conectados. O padrão X10 usa infra-estrutura elétrica da casa, a linha de alimentação, para conectar os dispositivos, o UPnP usa o Wi-Fi em casa para conectar os dispositivos, mas o roteador precisa ser compatível com UPnP, e hoje em dia eles são. Outro padrão usa a tecnologia de comunicação específica e precisa de uma nova infra-estrutura para conectar os dispositivos. A ZigBee usado para automação residencial normalmente é configurada como uma rede mesh, mas cada dispositivo precisa de um rádio ZigBee para ser capaz de se conectar à nova infra-estrutura de rede implantada para a casa.

A comparação destaca que essas tecnologias têm características diferentes. Dependendo do tipo, complexidade e tamanho do dispositivo ou sistema, uma tecnologia específica pode tornar-se mais adequado para o caso de uso estudada. No entanto, o foco em automação residencial cria limites e restrições para essas tecnologias, e, apesar de uma ampla gama de tecnologias, elas ainda não são capazes de atender a todas essas características restritivas.

O primeiro e segundo critérios, comunicação e infraestrutura, mostra duas características das normas que estão conectados. O padrão X10 utiliza a infraestrutura de rede elétrica da residência para conectar os dispositivos, enquanto o UPnP utiliza a infraestrutura de rede Wi-Fi, porém nesse caso o roteador precisa ser compatível com o UPnP, a partir dos anos 2000 a maioria está sendo produzida suportando este padrão. Outros padrões utilizam tecnologias de comunicação específicas e precisam de uma infraestrutura nova para conectar seus dispositivos. O padrão ZigBee utilizado para automação residencial é configurado em rede *mesh*, porém cada dispositivo precisa ter um rádio ZigBee para poder se conectar à essa nova infraestrutura de rede que foi criada para a residência.

A necessidade por uma nova infraestrutura aumenta o custo de instalação da solução de automação residencial. No entanto, o custo total pode ser afetado pela complexidade do padrão e pela necessidade de um maior poder de processamento, isto é, soluções mais simples necessitam de um menor poder de processamento e tendem a ser mais econômicas. A taxa de transferência de dados também afeta a

necessidade por poder de processamento e por quantidade de memória, quanto maior for essa necessidade de um hardware melhor e mais poderoso, maior será o consumo de energia e mais cara essa solução pode se tornar.

O nível abstrato do sistema pode afetar a quantidade de tipos de dispositivos que podem ser integrados pelo padrão, mas também podem afetar a eficiência da comunicação, as restrições no processamento, o consumo de energia e pode ser capaz de aumentar a complexidade e o custo de um sistema baseado no padrão relacionado. Todos esses critérios e outros requisitos estão conectados e cada padrão possui seu equilíbrio interno nessa relação. Sistemas diferentes de automação residencial possuem diferentes requisitos e utilizam o padrão que melhor satisfaz essas especificidades.

2.3.4.2 Resumo da comparação

Concentrando-se na automação residencial, alguns dos requisitos, tais como flexibilidade e modularidade, pode ajudar a aumentar as inserções dessas tecnologias do mercado. Portanto, o uso da infraestrutura existente para transferir qualquer tipo de dados com menor custo, complexidade e consumo de energia e maior poder de processamento, taxa de dados, segurança, privacidade e nível de abstração.

A última linha da Tabela 2 mostra o resumo das características mais desejadas e essas características desejadas são destacadas em negrito na tabela para cada tecnologia. A última coluna é a contagem de características desejadas que cada tecnologia tem, pela qual a tabela foi ordenada. Assim, de acordo com a tabela criada as tecnologias mais adequados para automação doméstica estão no topo da tabela.

Algumas das características desejadas são conflitantes, e os pesquisadores de cada tecnologia tem de escolher a forma de nivelar esses requisitos. Desta forma, eles podem chegar a uma solução final que pode entrar com sucesso no mercado-alvo. Para automação residencial, a flexibilidade, fácil integração de diferentes marcas, fácil instalação e manutenção, baixo custo devem ser os principais requisitos para os pesquisadores direcionarem suas pesquisas para facilitar a adopção dos sistemas de automação residencial.

2.3.4.3 Comentários sobre as tecnologias

Diversos novos sistemas para automação residencial e novos dispositivos para facilitar a vida doméstica são criados diariamente, tornando-os cada vez mais populares. Entretanto, o que deve ser destacado é que a infraestrutura HAN é apenas um dos aspectos que devem ser considerados (VAXEVANAKIS; ZAHARIADIS; VOGIATZIS, 2003). O sucesso na adoção de um dos padrões será baseado não apenas nessa integração, mas na efetividade, utilidade, flexibilidade e custo do sistema como um todo.

A adoção massiva da automação residencial poderá ser tornar uma realidade quando novos serviços forem capazes de atrair consumidores, fornecendo a funcionalidade, a flexibilidade e a qualidade apropriadas, reunindo os requisitos dos usuários, possuindo conteúdo suficiente, e se tronar melhor em comparação aos sistemas autônomos não integrados existentes (VAXEVANAKIS; ZAHARIADIS; VOGIATZIS, 2003).

De acordo com (BALTA-OZKAN et al., 2013) existem três métodos para que a automação residencial se propague:

- *Retrofitting*: aperfeiçoando as residências existentes, adicionando a elas novas funcionalidades;
- Convertendo outras propriedades, adaptando-as às necessidades dos compradores;
- Residências construídas com esse objetivo.

Os últimos dois são mais prováveis de serem usados devido à otimização do custo. Entretanto, as novas tecnologias estão sendo trazidas para dentro de residência em partes, divididas em pequenos pedaços (BALTA-OZKAN et al., 2013). Esse desenvolvimento de uma tecnologia modular, de custo menor e acessível é responsável por essa incorporação gradual dentro de residências novas e também de residências já existentes (WILSON; HARGREAVES; HAUXWELL-BALDWIN, 2015).

A *smart home* pode ser considerada um dos domínios primários de aplicação IoT (TUCIC et al., 2014). A evolução desse mercado pode ser dividida em duas fases (BALTA-OZKAN et al., 2013):

1. Equipamentos autônomos e independentes;

2. Equipamentos inteligentes com conectividade e integração com a rede;

De acordo com um especialista em (BALTA-OZKAN et al., 2013), apesar da importância de um padrão universal para superar o problema de interoperabilidade, um caminho que esse mercado pode seguir é aquele em que os vencedores dessa evolução da *smart home* tenderão a definir padrões para interoperabilidade (BALTA-OZKAN et al., 2013). Essa unificação resulta em um sistema coerente e na fácil distribuição da informação (TUCIC et al., 2014).

Entretanto, com o advento das tecnologias inteligentes, as *smart homes* se tornarão algo comum na vida das pessoas, mesmo que eles não queiram (WILSON; HARGREAVES; HAUXWELL-BALDWIN, 2015).

2.4 DESAFIOS EM ABERTO E DIRECIONAMENTOS FUTUROS

Essa seção apresenta os desafios e expectativas para o futuro. A partir das revisões apresentadas, é possível descobrir desafios em aberto e tendências para soluções futuras, e utilizá-los como diretrizes para próximas pesquisas.

Desde o início dos anos 2000, o mercado de automação residencial não está unificado, não existe uma tecnologia adotada universalmente, e a heterogeneidade nesse cenário é um desafio para a integração e consolidação desse mercado. A primeira ideia para suplantar esses problemas foi criar conexões entre tecnologias diferentes, e cada nova tecnologia que surge deve ser adicionada nessa ponte, entretanto, a necessidade de se ter um nó central para controlar o sistema pode limitar a funcionalidade das tecnologias conectadas. A segunda ideia foi criar um protocolo comum com o qual todos os dispositivos seriam compatíveis (FERREIRA; CANEDO; SOUSA JUNIOR, 2013). Esse protocolo necessitaria ser muito flexível e robusto para conseguir integrar todos os tipos de dispositivos, porém isso significa que todos os dispositivos devem ser capazes de processar esse único protocolo. Uma terceira ideia seria criar uma arquitetura dividida em duas camadas: a primeira para integrar os subsistemas de automação da residência, e a segunda na qual cada subsistema utiliza uma tecnologia de comunicação específica para conectar seus dispositivos.

2.4.1 Questões em discussão

Eletrodomésticos estão se tornando equipamentos baseados na Tecnologia da Informação e dispositivos básicos de uso doméstico estão se tornando mais inteligentes. Eles não são mais equipamentos independentes, pois estão conectados à uma rede para trocar informações. Entretanto, existem diferentes padrões, tecnologias e soluções para integrar esses dispositivos, porém nenhum ainda foi adotado universalmente. Diferentes fornecedores e associações vêm desenvolvendo diferentes tecnologias e publicando diferentes padrões para diversos propósitos, porém, estes não trabalham juntos sem uma "ponte" entre eles (NGO, 2007).

De acordo com Chen (2014), a heterogeneidade traz desafios específicos que precisam ser abordados. Igualmente, a comunicação de objetos heterogêneos na rede M2M de uma residência é um grande problema pois diferentes objetos fornecem diversas informações em distintos formatos para vários objetivos (KIM; YOUM, 2013; KIM et al., 2014). Soluções diferentes utilizando a mesma tecnologia de rede podem ainda ser incompatíveis pois mesmo utilizando o mesmo protocolo de comunicação, os objetos e interfaces não são padronizados e a linguagem de um se torna incompreensível para o outro. Portanto, as próprias soluções podem se tornar incompatíveis.

Para controlar remotamente um grande número de diferentes equipamentos M2M e alcançar a interoperabilidade entre eles de uma maneira efetiva e segura, é possível interconectar componentes não baseados em IP e equipamentos com redes externas baseadas em IP (CHEN et al., 2014; KIM; YOUM, 2013; KIM et al., 2014). De acordo com Chen et al. (2014), os esquemas convencionais para realizar essa conexão ainda são, entretanto, insuficientes para alcançar transmissões confiáveis, robustas e com pouco atraso.

Outra abordagem utilizada é conectar cada equipamento diretamente a uma rede IP. Essa pode ser uma solução possível para dispositivos com mais energia para processar o protocolo TCP/IP. Entretanto, para equipamentos menores, com recursos ou requisitos limitados, como baixo consumo de energia ou consumo, essa solução é inviável.

O mercado de sistemas residenciais não é grande e desenvolvido suficiente para suportar essa grande variedade e granularidade das soluções propostas para

interoperabilidade de M2M (WACKS, 2002). Algumas dessas soluções foram criadas com objetivos diferentes, e agora estão sendo utilizadas para interconectar equipamentos heterogêneos, criando, portanto, sobrecarga desnecessária, brechas de segurança e outros problemas derivados dessa adaptação (KIM; YOUM, 2013; KIM et al., 2014). Além disso, a propagação da adoção cria um ambiente não integrado em que inúmeras soluções finais de interoperabilidade não são compatíveis umas com as outras. Mas há várias pequenas soluções e projetos para interligar uma classe de dispositivos que poderiam ser utilizados como um subsistema em uma infraestrutura de comunicação mais ampla.

Essas diversas pequenas soluções e projetos para interconectar uma classe de dispositivos podem ser utilizadas como um subsistema em uma solução de infraestrutura de comunicação mais ampla. Outro fator que impede o crescimento do mercado é o custo. Empresas de construção não estão dispostas a adicionar tecnologias de *smart home* ao projeto da construção e ainda menos dispostas a instalá-las. Residências e edifícios deixam, assim, de contar com uma infraestrutura para automação residencial, e mesmo com a habilidade de acomodar essas tecnologias (BALTA-OZKAN et al., 2013).

2.4.2 Futuro

A literatura destaca duas formas de obter consistência e coerência com um sistema de *smart home* (BALTA-OZKAN et al., 2013), e as soluções apresentadas podem ser qualificadas como:

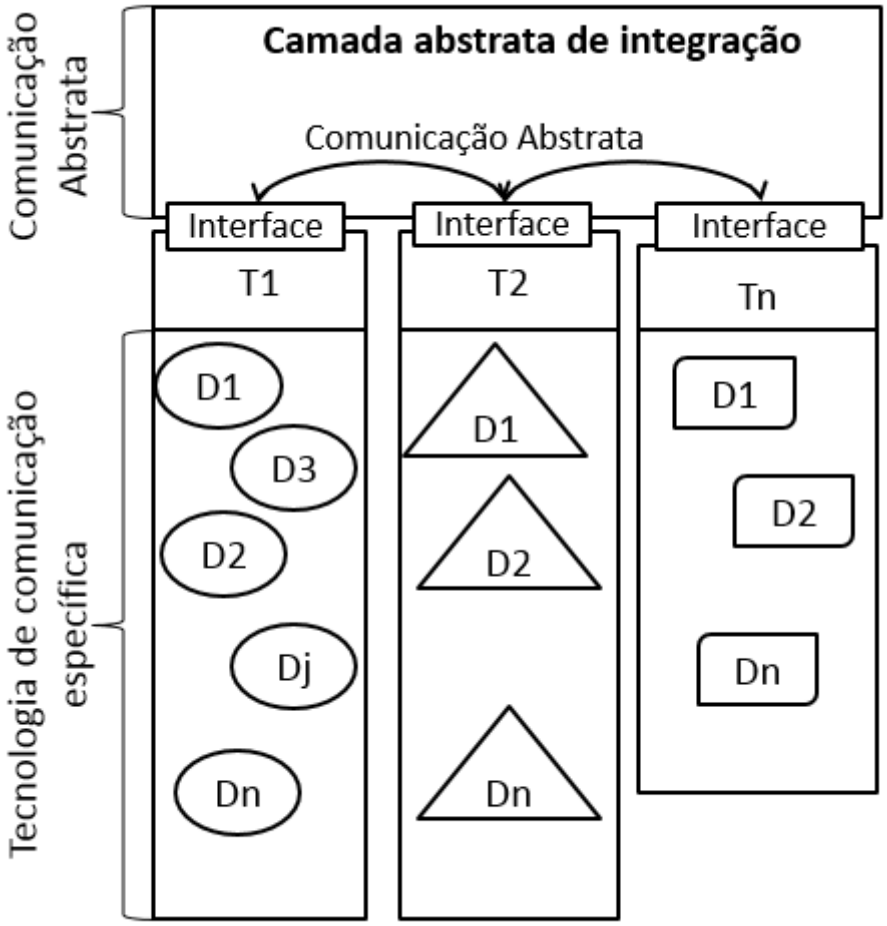
- A primeira seria um padrão universal para protocolo de comunicação. Diversos padrões têm sido criados por organizações privadas, associações, comitês, alianças e cooperativas de empresas. Mas nenhum foi adotado universalmente e se tornou um padrão para *smart homes* ou automação residencial.
- A segunda seria um *gateway* central de rede, em que esses gateways conectam as sub-redes, traduzindo seus protocolos. Entretanto, as demandas futuras para *smart homes* são imprevisíveis e por isso os *gateways* necessitam ser altamente sofisticados e flexíveis para procurar, reconhecer e

se adaptar a todos os novos equipamentos com novos protocolos e adicioná-los à rede.

Uma terceira abordagem pode ser considerada como uma combinação dessas duas. Dessa forma pode-se usar um protocolo de comunicação padronizado como um *gateway* de integração entre diferentes subsistemas. Assim, uma camada abstrata e flexível é criada no topo das outras tecnologias de equipamentos de comunicação. Essa camada acima é utilizada para integrar e interconectar as outras sub-redes. Por essa razão, ela necessita ser abstrata e flexível o suficiente para suportar a intercomunicação entre todos os subsistemas de forma automática e transparente para o usuário.

Os subsistemas podem usar tecnologias, protocolos e arquiteturas diferentes para conectar seus dispositivos em suas sub-redes. Para construir esse subsistema, é possível escolher a tecnologia que melhor atende os requisitos do tipo de equipamentos utilizados no subsistema. A Figura 11 apresenta o exemplo de arquitetura dessa terceira futura abordagem.

Figura 11 - Possível arquitetura futura



Fonte: autor

Os subsistemas podem ser construídos por diferentes fabricantes, utilizar arquitetura centralizada ou distribuída, ter um *middleware* para orquestrar os dispositivos ou ter equipamentos autônomos. Eles podem estar conectados através da rede elétrica, de uma rede ZigBee ou qualquer outra tecnologia de comunicação. Isso dependerá das exigências dos equipamentos do subsistema que utilizam essa sub-infraestrutura. Em resumo, o subsistema pode ser tão diferente quanto precise ser, desde que possua a interface da camada de controle e comunicação abstrata acima.

A camada abstrata pode utilizar uma tecnologia flexível e abstrata. Ela precisa facilitar a integração dos subsistemas para que seja fácil de utilizar e de manter. Suas características precisam aumentar e promover a adoção do sistema pelo

usuário final. Essa adoção massiva de sistemas de integração e automação pelos usuários finais é importante para fomentar a livre competição de mercado.

Compondo o sistema final por essas duas camadas, uma de controle no topo e outra especializada na base, é possível dar ao sistema as melhores qualidades das duas camadas. O sistema pode se tornar flexível, integrado e interconectado, e simultaneamente especializado, eficiente e leve para melhor se ajustar em todos os requisitos dos equipamentos.

Com essa arquitetura, o sistema pode ser composto por uma camada de integração e vários subsistemas de automação. Esses subsistemas podem ser construídos por diferentes fabricantes, vendidos individualmente ou como um pacote de diferentes fornecedores. Essas características permitem que os consumidores escolham os sistemas que melhor se encaixam às suas necessidades e podem ser facilmente integradas a ele.

Essa flexibilidade do mercado facilita a aquisição, uso e manutenção do sistema, e espera-se que ajude a aumentar a adoção dos mesmos. A variedade de fabricantes produzindo subsistemas pode criar um ambiente de livre competição que pode diminuir os altos preços desse tipo de sistema, e além disso aumentar a adoção pelo usuário final.

2.5 FECHAMENTO

Esse capítulo analisou algumas tecnologias de rede para a infraestrutura residencial, utilizados para automação e integração de tecnologias inteligentes, com o objetivo de mostrar tendências comuns em uma *smart home*. Para construir uma *smart home*, essa infraestrutura precisa integrar os diferentes sistemas e dispositivos de automação para garantir que funcionem juntos como um sistema inteligente unificado. Infelizmente, para o usuário final, nenhuma dessas tecnologias está sendo adotada como a principal tendência do mercado.

Essa heterogeneidade está dificultando o crescimento e evolução desse mercado. Para chegar a um cenário no qual o usuário terá uma residência completamente conectada e integrada, um padrão deve se sobressair e se tornar o líder do mercado. Dessa forma todos os equipamentos e eletrodomésticos serão inteligentes e capazes de trocar informações entre eles. A residência terá uma

inteligência artificial central que será capaz de orquestrar e administrar a residência como uma entidade única para melhorar a vida de seus moradores.

A visão do futuro pode apenas se materializar se um padrão se destacar e for adotado em todo o mundo. Competição de mercado natural pode acontecer, e diversos fabricantes de equipamentos e eletrodomésticos podem fabricar seus produtos para serem escolhidos pelo usuário e integrados facilmente no ambiente da residência. Esse ambiente residencial deve suportar diferentes marcas de equipamentos trabalhando juntos.

Cada uma das tecnologias analisadas tem o potencial para se tornar o padrão adotado. Todas elas possuem pontos fortes e pontos fracos, vantagens e desvantagens, mas a vencedora dessa competição se tornará a tecnologia padrão e guiará as tendências do mercado até que, com a dinamicidade do mercado, um padrão evolua e force os fabricantes a produzir aparelhos seguindo-o.

3. MATERIAIS E MÉTODOS

A metodologia utilizada no presente trabalho é caracterizada como exploratória e apresentada por Gil (2010). Nesta modalidade, uma análise teórica aprofundada é realizada, a partir da qual é definida a arquitetura a ser utilizada e elaborado um modelo teórico. Posteriormente, é feito um estudo experimental, no qual são realizadas simulações do sistema visando avaliar o comportamento das variáveis e extrair conclusões que permitam avaliar o sistema em relação ao original.

A aplicação dessa metodologia neste trabalho produziu cinco etapas, que serão descritas em profundidade a seguir:

1) Escolha do padrão;
2) Elaboração de um modelo estendido do sistema;
3) Definição da arquitetura;
4) Simulação computacional do sistema;
5) Avaliação dos resultados.

3.1 ESCOLHA DO PADRÃO

Com a grande variedade de soluções heterogêneas no contexto de automação residencial, essa etapa consistiu na escolha de um padrão ou tecnologia para servir como base para a elaboração do modelo estendido. A escolha da tecnologia foi realizada a partir da análise e comparação, feitas durante o levantamento do estado da arte, de padrões e tecnologias de automação e interoperatividade.

3.2 ELABORAÇÃO DO MODELO ESTENDIDO

Esta etapa consistiu na extensão do modelo do padrão UPnP, com foco na automação residencial. Este padrão é composto por uma série de descrições padronizadas dos dispositivos. Porém, essa descrição dos dispositivos voltados para a automação residencial foi pouco explorada ainda. Poucos tipos de dispositivos de automação foram inclusos, e esses poucos dispositivos são simples e de baixa complexidade.

O objetivo desta etapa é a elaboração de uma extensão do modelo simples de dispositivo de automação para um modelo mais complexo. Este novo modelo tem

como objetivo possibilitar a descrição de subsistemas de automação, utilizando a estrutura de alto nível do UPnP de modo mais eficiente.

Esses subsistemas passaram a atuar na estrutura UPnP como dispositivos complexos, porém na realidade são sistemas compostos de um ou mais dispositivos que podem utilizar diferentes meios físicos, protocolos e outros padrões para se comunicarem internamente. Essa nova estrutura possibilita o controle dos subsistemas utilizando uma linguagem abstrata, enquanto os dispositivos que fazem parte do subsistema são controlados e integrados utilizando uma tecnologia apropriada para o tipo de comunicação necessária.

Um exemplo seria um subsistema de iluminação no qual todas as lâmpadas do tipo liga/desliga estão conectadas, e assim pode-se utilizar uma tecnologia de comunicação mais simples e lenta para baratear o custo de cada lâmpada. Enquanto, um subsistema de monitoramento por vídeo a tecnologia mais adequada deve ser uma que possibilite a transferência de vídeo em alta qualidade em alta velocidade. Porém, ambos os subsistemas podem ser integrados ao sistema de automação da residência através do UPnP, os controladores dos subsistemas estão conectados à rede da residência, na qual a estrutura de automação está instalada, e devem possuir uma interface UPnP que disponibiliza o controle em alto nível do subsistema.

3.3 DEFINIÇÃO DA ARQUITETURA

A extensão do modelo e a definição da arquitetura são trabalhos complementares, pois a definição da arquitetura mais adequada depende de como o modelo funciona. Por esse motivo, a extensão do modelo UPnP e a definição da arquitetura, focados na automação residencial, foram realizados em paralelo.

Essa arquitetura é composta por duas camadas. Uma camada que utiliza o UPnP estendido para o controle e integração entre subsistemas da residência, e outra que utiliza a tecnologia de comunicação dedicada que melhor atende os requisitos dos dispositivos do subsistema. Os subsistemas podem utilizar tecnologias distintas para sua comunicação interna, porém são integrados pela camada de controle do UPnP estendido.

3.4 SIMULAÇÃO COMPUTACIONAL DO MODELO ESTENDIDO

Esta etapa consistiu na escolha de um caso de uso, sua construção utilizando o modelo estendido e sua simulação. A partir desse objetivo criou-se um ambiente virtual com um subsistema composto de diversos dispositivos que pode atuar simultaneamente. Esse ambiente virtual consiste em clientes UPnP que controlam dispositivos virtuais, que não tem atuação física, com interface gráfica para visualização de suas ações que são registradas em arquivo texto.

Para criar esse ambiente pode-se utilizar uma ferramenta de desenvolvimento de dispositivos para o padrão UPnP (KIM et al., 2002), muitas dessas ferramentas são pagas e outras abertas, *open source*, (UPNP FORUM SDKS, 2014). Os kits, listados no site no UPnP Forum [2], fornecem blocos de construção para o desenvolvimento de dispositivos e serviços UPnP. No escopo deste trabalho foi mais adequado a utilização de uma ferramenta de uso livre e flexível, que permitiu a criação simplificada de dispositivos e subsistemas a partir de modelos estendidos do padrão UPnP, para que depois pudessem ser realizados testes rápidos e iterativos sobre o modelo criado.

O *framework Developer Tools for UPnP Technologies* (OPEN SOFTWARE PROJECTS, 2014) foi escolhido para ser utilizado na criação dos dispositivos virtuais. Essa ferramenta, patrocinada e suportada pela Intel, tem sua documentação disponibilizada na sua '*Developer Zone'* (INTEL UPNP TOOLS, 2014).

Com essa ferramenta foi possível obter, a partir da descrição UPnP, um modelo de código do cliente UPnP utilizado na criação do dispositivo (INTEL DEVELOPER TOOL, 2014). Com isso pôde-se criar clientes complexos que controlam um ou mais dispositivos virtuais para testar a expansão do modelo UPnP realizada na etapa anterior (INTEL UPNP EXAMPLE, 2014).

A simulação consistiu em um subsistema multidispositivo construído a partir do modelo UPnP estendido com o qual os clientes UPnP puderam interagir e controlar os diversos dispositivos. Além disso, pôde-se criar mais de uma instância desse subsistema, e dessa forma simular a comunicação e a integração entre os subsistemas virtuais criados de automação residencial.

---

2 www.upnp.org/

Esses testes foram elaborados de modo a verificar a consistência do modelo e a interação entre subsistemas gerada a partir desse modelo.

3.5 AVALIAÇÃO DOS RESULTADOS E DISCUSSÕES

Após a simulação computacional do modelo UPnP estendido, a avaliação e análise dos resultados foi realizada qualitativamente, e os pontos positivos e negativos resumidos utilizando a técnica SWOT (*Strengths*, *Weaknesses*, *Opportunities* e *Threats*), de forma a realizar uma comparação do modelo estendido com as soluções existente. Em sequência, pontos de discussões adicionais foram apresentados.

3.6 FECHAMENTO

Neste capítulo foram apresentados os matérias e métodos utilizados na realização deste trabalho. Em seguida será apresentada a proposta de extensão do modelo UPnP.

4. PROPOSTA DE EXTENSÃO DO MODELO E DEFINIÇÃO DA ARQUITETURA

Este capítulo apresenta o modelo de descrição de dispositivos do modelo UPnP original, e sua extensão. Após descrever o modelo UPnP estendido, foi realizada uma análise teórica dos impactos dessa extensão.

4.1 ESCOLHA DO UPNP

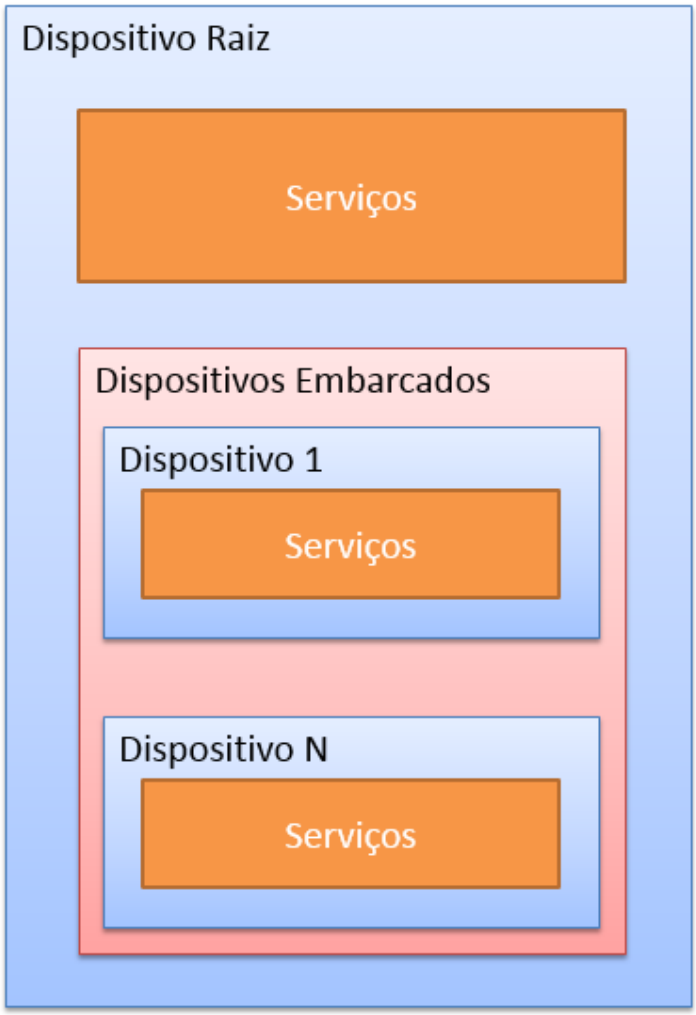
O padrão UPnP foi apresentado e discutido na revisão de literatura anteriormente. Ele foi escolhido devido a sua flexibilidade, e por ele já ser utilizado no ambiente residencial. Com o padrão UPnP escolhido pôde-se estendê-lo para melhor adequá-lo à automação residencial.

4.2 O MODELO UPNP

O padrão UPnP foi desenvolvido para integrar dispositivos unitários, por exemplo, possibilitar que uma impressora fique disponível para todos os usuários em uma residência, sem a necessidade de instalação do *drivers* especifico dessa impressora (MILLER et al., 2001). Neste padrão, a descrição de dispositivos é realizada por meio de um arquivo XML. Quando um cliente UPnP requisita a descrição de um dispositivo, esse arquivo XML é enviado ao cliente, que interpreta e permite conectar e controlar os dispositivos.

O modelo de descrição de um dispositivo UPnP, padronizada na ISO/IEC29341, segue a estrutura na Figura 12. Essa descrição do dispositivo é composta pela descrição de seus serviços e dispositivos embarcados. A descrição desses dispositivos segue o mesmo padrão, e assim recursivamente (KIM et al., 2002).

Figura 12 - Estrutura de descrição de dispositivos UPnP



Fonte: elaborado pelo autor com base na norma ISO/IEC29341

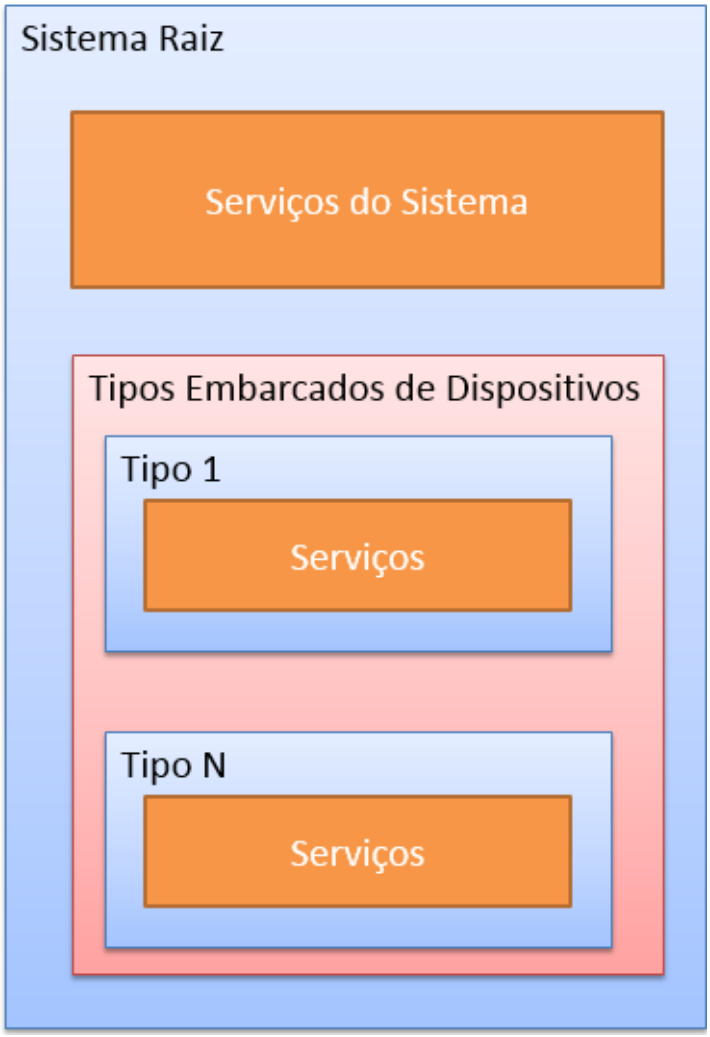
O dispositivo raiz pode ter tantos dispositivos embarcados quanto forem necessários, sendo que todos devem estar descritos por completo em seu bloco ou *tag*. Se o dispositivo raiz possuir dispositivos embarcados iguais, a mesma descrição será repetida uma vez para cada instância de dispositivo embarcado presente no dispositivo raiz.

4.3 PROPOSTA DE EXTENSÃO DO MODELO UPNP

Utilizando a mesma estrutura do arquivo de descrição de dispositivos no padrão UPnP, e alterando-se o significado de alguns blocos, pode-se extender o modelo

UPnP para se obter um modelo estendido de sistema multidispositivos, representado da Figura 13.

Figura 13 - Modelo estendido para sistemas multidispositivos



Fonte: elaborado pelo autor com base na norma ISO/IEC29341

A descrição em Tipo, contido no bloco Tipos Embarcados de Dispositivos, segue a mesma estrutura da descrição anterior em Dispositivo, mas neste uma categoria de dispositivo é descrita ao invés de um dispositivo único.

As mudanças sintáticas dentro do contexto de automação residencial proporcionam as seguintes vantagens e desvantagens:

1. Dispositivo Raiz -> Sistema Raiz

   Descrição do sistema ao invés de um único dispositivo.

2. Serviços -> Serviços do Sistema

Os serviços do sistema devem ser utilizados para obter as informações sobre os dispositivos do sistema, como o número de dispositivos de cada tipo, sua localização, etc., além de serviços de controle de múltiplos dispositivos simultaneamente ou serviços de controle do sistema.

3. Dispositivos Embarcados -> Tipos Embarcados de Dispositivos

O grupo de tipos de dispositivos embarcados do sistema permite a economia de tamanho de descrição para sistemas com múltiplos dispositivos iguais, dessa maneira um sistema com um ou mais dispositivos iguais terá uma descrição de tipo de dispositivo, ou seja, uma descrição por tipo de dispositivo embarcado.

4. Dispositivos [Embarcados] -> Tipos [Embarcados]

A descrição do tipo de dispositivo segue a mesma estrutura da descrição de um dispositivo, porém não especifica um dispositivo unicamente, mas sim um grupo de dispositivo iguais. Dessa maneira para um cliente conseguir controlar um dispositivo precisará obter a identificação dos dispositivos do sistema através dos serviços de sistema e utilizar a descrição do tipo de dispositivo para acessar o dispositivo individualmente.

4.4 PROPOSTA DA ARQUITETURA DO SISTEMA DE AUTOMAÇÃO

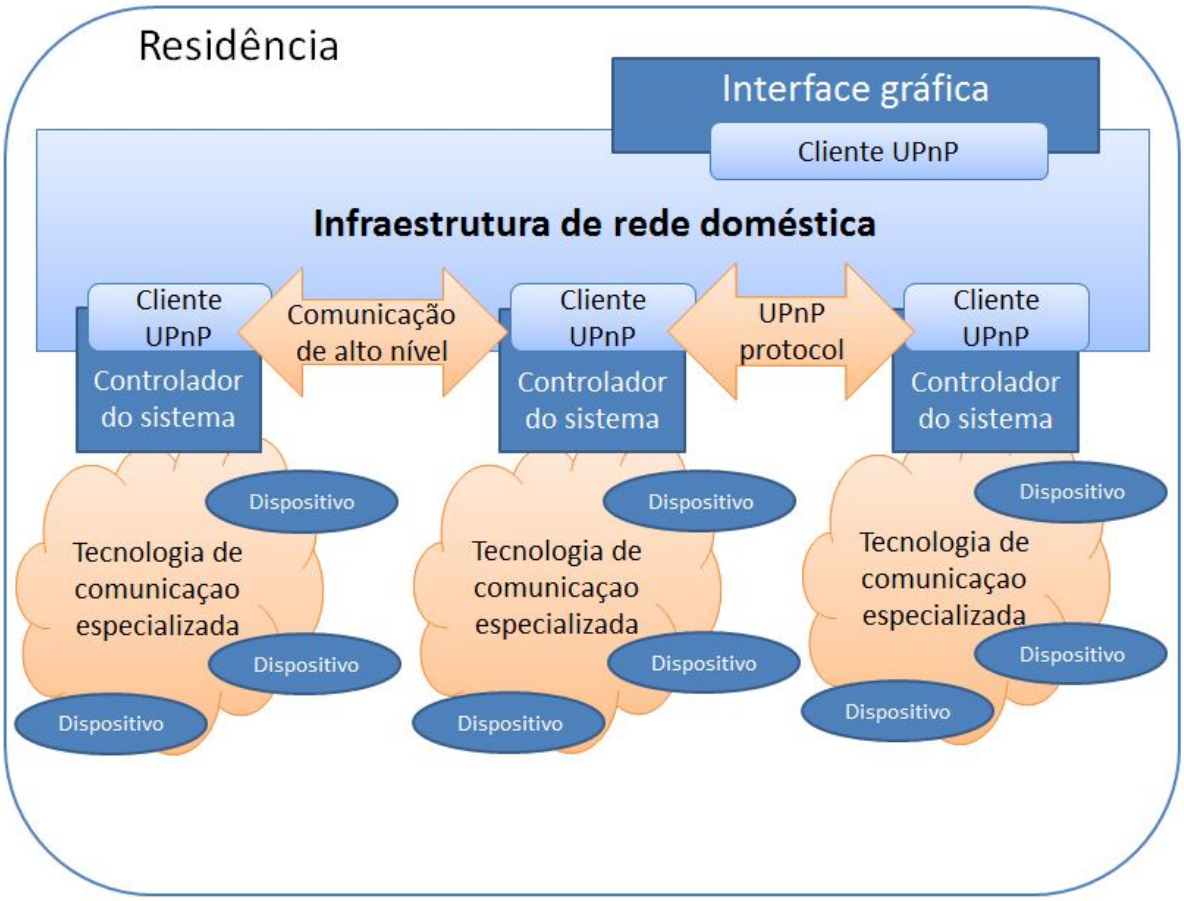
A extensão proposta do modelo UPnP tem por objetivo aumentar a flexibilidade do padrão UPnP para adaptá-lo a subsistemas multidispositivos de automação residencial, propiciando e favorecendo uma arquitetura com duas camadas. A primeira camada é abstrata e tem a função de controle e integração dos subsistemas, já a segunda é especializada e tem a função de comunicação entre os dispositivos do subsistema.

Utilizando-se essa extensão, a comunicação integrada entre os diversos subsistemas de automação acontece de forma transparente e integrada nessa camada abstrata. A camada abstrata segue o padrão UPnP estendido, assim o

controlador de cada subsistema se conecta nesta camada para interagir com outros subsistemas através de um cliente UPnP.

Cada subsistema pode fazer a comunicação e o gerenciamento interno entre seus dispositivos através de uma tecnologia de comunicação otimizada para o tipo de dispositivo do sistema. Por exemplo, um subsistema de iluminação pode ter lâmpadas com comando remoto sem fio, e utilizar o ZigBee em rede *mesh* para a comunicação entre elas, devido ao seu baixo custo e consumo, e mesmo assim manter a cobertura em toda a residência devido à distribuição das lâmpadas. Dessa forma, a central de controle do subsistema deve ter uma interface UPnP que pode ser acessada por qualquer cliente UPnP para controlar individualmente ou em grupo essas lâmpadas. A Figura 14 apresenta a arquitetura de um sistema de automação que utilize o modelo UPnP estendido.

Figura 14 - Arquitetura proposta para o modelo estendido



Fonte: autor

Essa interface UPnP consegue fornecer uma linguagem de controle abstrata, genérica e ampla que poderá ser utilizada diretamente pelos moradores ou por outros subsistemas da residência. No exemplo anterior os moradores podem

controlar uma ou múltiplas lâmpadas simultaneamente através de um cliente UPnP, ou o subsistema de segurança da residência pode acender e apagar lâmpadas para simular algum morador na residência enquanto todos estão viajando.

Por fim, essa arquitetura possibilita um controle descentralizado de alto nível e um controle centralizado em cada subsistema, o qual permite a adição de funcionalidades e inteligência ao subsistema, devido à possibilidade de controlar diversos dispositivos simultaneamente. Dessa forma, esta proposta pode aumentar o valor agregado do subsistema, enquanto a modularização do sistema de automação da residência permite que o usuário componha seu sistema de automação residencial por diferentes módulos, ou seja, subsistemas, de diferentes fabricantes, com diferentes características, de modo a melhor satisfazer suas necessidades.

4.5 ANÁLISE DA PROPOSTA

O modelo estendido apresentado utiliza o paradigma de interface, sendo que os tipos de dispositivos embarcados presentes na descrição do subsistema são as interfaces as quais os dispositivos são instanciados. Para cada tipo de dispositivo, o subsistema pode ter um ou mais dispositivos instanciados presentes no subsistema.

Esse agrupamento de dispositivos possibilita, inicialmente, a simplificação na descrição do sistema, com uma única descrição por tipo de dispositivo embarcado. Outra vantagem de se agrupar dispositivos em subsistemas é a possibilidade de controlar todos os dispositivos simultaneamente, através de serviços do sistema. Esses subsistemas podem ter quantas funcionalidades em formato de serviços quanto o fabricante desejar conceber, para facilitar sua utilização e agregar valor ao seu sistema.

O modelo UPnP estendido também traz características que podem aumentar a complexidade do sistema. Uma delas é o aumento da complexidade do cliente UPnP que irá utilizar esses subsistemas. Este cliente deve obter a lista de dispositivos e seus respectivos tipos através de um serviço do subsistema para assim utilizar a interface do tipo específico para controlar cada dispositivo individualmente criando, assim, a necessidade de mapear cada dispositivo à sua respectiva interface. Essa lista de mapeamento de dispositivos pode ser uma lista simples com seus respectivos tipos, ou pode também conter mais informações sobre

cada dispositivo, como sua localização na residência, alguma restrição ou deficiência que o dispositivo pode ter, e meta-informações que podem ajudar o cliente UPnP a apresentar o controle da residência ao usuário com mais clareza e facilidade de uso.

Uma característica importante de sistema de automação residencial que segue esse modelo de divisão em subsistemas é a separação da comunicação em dois níveis de abstração: um (i) nível abstrato no qual o UPnP estendido é utilizado para integrar os distintos subsistemas, e outro (ii) nível concreto, de baixo nível, que utiliza uma linguagem mais próxima aos dispositivos, específica ao subsistema, para conectar os dispositivos que pertencem à um subsistema. Dessa forma pode-se separar as redes e sub-redes, possibilitando a utilização de diferentes tecnologias. Na camada do UPnP utiliza-se a rede IP da residência, e nas redes dos subsistemas os fabricantes podem escolher a tecnologia que melhor atenda às necessidades e aos requisitos de cada sistema.

No entanto, esse formato de sistema também traz desvantagens. Essa modularização do sistema de automação residencial em subsistema cria a necessidade de uma camada de integração que seja capaz e flexível suficiente para possibilitar que qualquer tipo de subsistema consiga se conectar e interagir dentro do ambiente da residência de modo transparente. Assim, cada subsistema é obrigado a ter uma interface de controle seguindo esse padrão. O padrão UPnP apresenta essas características, vantagens e flexibilidade, além da penetração de mercado já presente, e por isso foi escolhido para ser estendido. A extensão do padrão conserva essas características e propõe o aumento de flexibilidade e novas funcionalidades focadas no ambiente residencial.

A modularização do sistema como proposto possibilita a integração de módulos, isto é, subsistemas, de diferentes fabricantes, que podem concorrer com subsistemas de diferentes segmentos. Assim o usuário pode escolher e comprar o subsistema que lhe agrade mais para montar seu sistema de automação residencial personalizado sem a necessidade de adaptações e sem sua fidelização à um único padrão ou fabricante. Os usuários podem comprar produtos de prateleira e estes podem se integrar à automação de sua residência de modo automático e transparente. Essa modularização também possibilita a atualização simplificada do sistema de automação residencial através da troca de subsistemas ultrapassados

por novos, e da mesma forma pode-se adicionar subsistemas extras que possam surgir no futuro.

4.6 FECHAMENTO

Neste capítulo foi apresentada a proposta de pesquisa deste trabalho, a extensão do padrão UPnP, e como este foi estendido. A seguir será apresentado como o caso de uso foi construído para a realização de simulações do UPnP estendido.

5. SIMULAÇÃO COMPUTACIONAL DO MODELO PROPOSTO

Após a extensão do modelo do padrão UPnP, foi elaborado um caso de uso e implementado o seu subsistema. A partir deste foram realizadas simulações computacionais para testar essa extensão.

5.1 FERRAMENTAS DE IMPLEMENTAÇÃO E SIMULAÇÃO

A ferramenta *Developer Tools for UPnP* (DEVELOPER TOOLS, 2015) é um conjunto de softwares de desenvolvimento e teste de softwares de dispositivos compatível com o padrão UPnP. Este projeto é *open source* baseados na licença Apache 2.0. Essa ferramenta é capaz de gerar uma estrutura de código na Linguagem C, compatível com as camadas do protocolo UPnP, a partir da especificação de um modelo de dispositivo.

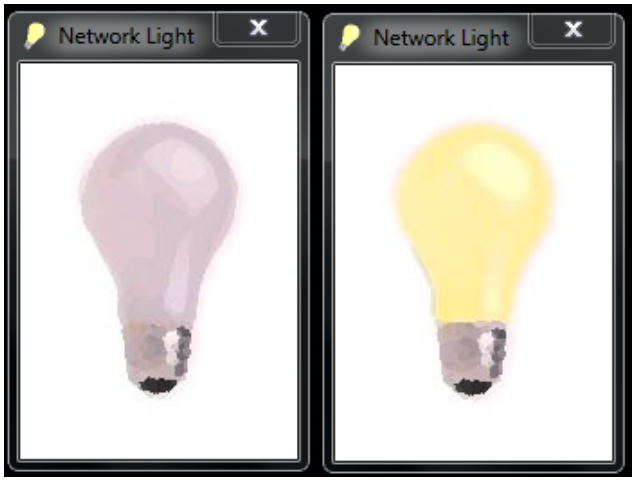
A primeira etapa desse processo é a criação dos serviços no *Service Author*, especificando-se as entradas e saídas dos serviços criados. A segunda etapa é a criação do dispositivo ou sistema no *Device Builder*, especificando o sistema raiz e seus tipos de dispositivos, para assim importar os serviços criados para as respectivas descrições dos tipos de dispositivos ou para o sistema raiz. Além disso, a partir desse modelo criado, o software *Device Builder* gera a estrutura de código.

Essa estrutura de código tem uma função para cada serviço do dispositivo criado. O desenvolvedor pode adicionar as ações para cada função correspondente ao respectivo serviço, e os códigos dessas ações são escritos em Linguagem C no corpo da respectiva função. Para isso pode-se utilizar qualquer software de edição de texto ou um software de programação específico. Depois de adicionar o código das funções, pode-se compila-lo e executar o programa do dispositivo criado.

A ferramenta também contém um cliente UPnP para que o usuário possa interagir e controlar o dispositivo criado, e um exemplo de dispositivo virtual executável com interface gráfica, para simular o dispositivo real. O dispositivo de exemplo é uma lâmpada, e em sua janela aparece uma imagem de sua representação. Quando o cliente UPnP envia um comando de acender, o dispositivo troca a imagem de uma lâmpada apagada por uma acesa, a Figura 15 apresenta

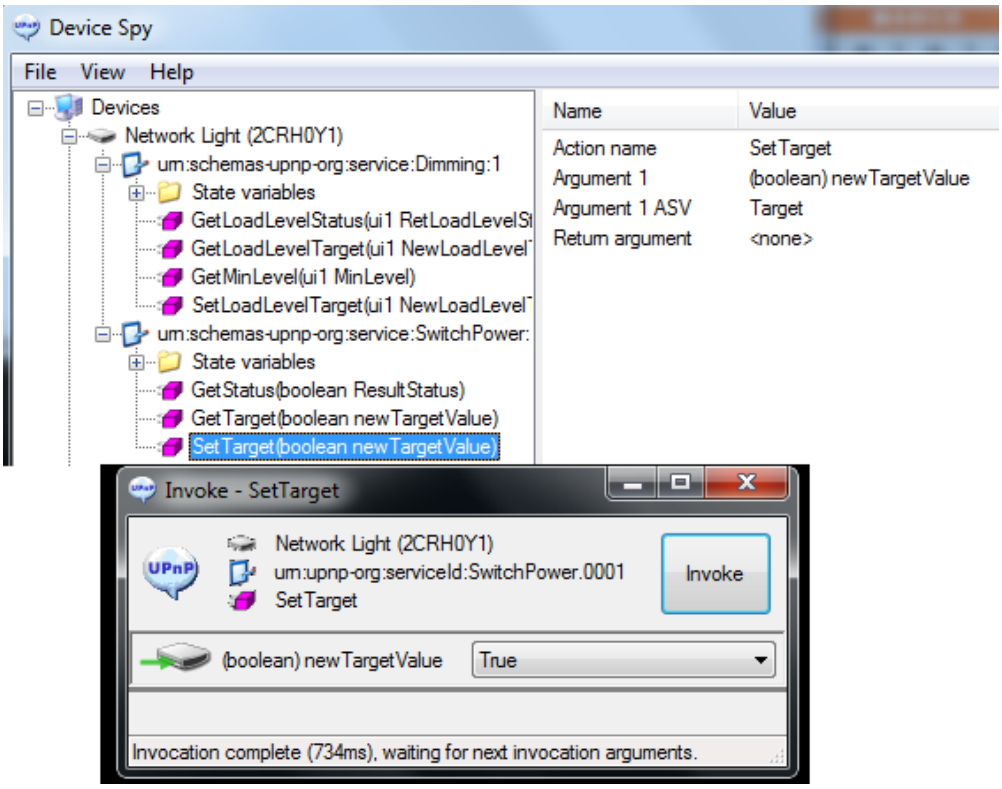
essa interface, enquanto a Figura 16 apresenta o cliente UPnP controlando esse dispositivo exemplo.

Figura 15 - Dispositivo exemplo



Fonte: Developer Tools for UPnP (2015)

Figura 16 - Cliente UPnP da ferramenta



Fonte: Developer Tools for UPnP (2015)

5.2 CASO DE USO PARA VALIDAÇÃO DO MODELO PROPOSTO

O caso de uso escolhido para fazer a validação e testes do modelo UPnP estendido foi um sistema de iluminação multidispositivo, composto por dois tipos distintos de lâmpadas. Este sistema foi desenvolvido para exemplificar um sistema de uso corriqueiro em uma residência, que pode ser considerado de médio porte e média complexidade. Assim, esse sistema de iluminação se torna um subsistema do sistema de automação residencial.

Esse sistema prevê dois tipos de lâmpadas:

- a simples, que permite apenas ligá-la e desligá-la;
- a dimerizável, que permite o controle de intensidade luminosa.

5.3 IMPLEMENTAÇÃO DO CASO DE USO

Neste item será descrita a implementação do caso de uso do subsistema de iluminação.

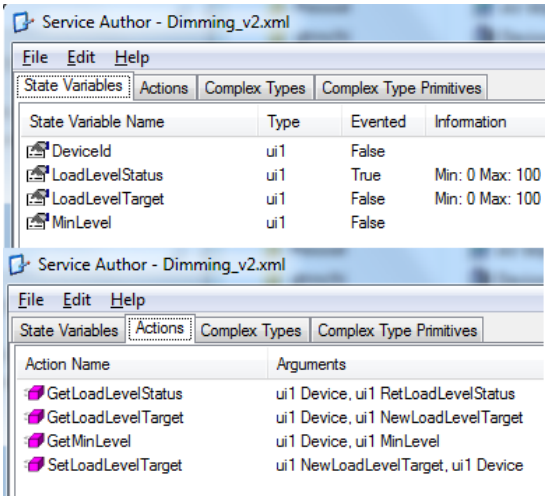
5.3.1 Criação do modelo do subsistema de iluminação

A criação do modelo do subsistema de iluminação foi realizada em duas etapas:

- Etapa 1: criação dos serviços
- Etapa 2: criação da descrição do sistema

Foram criados os serviços dos dispositivos para ligar e desligar as lâmpadas, e controlar sua intensidade. Nas Figura 17 e Figura 18 estão apresentadas as telas de variáveis e ações do *Service Author*:

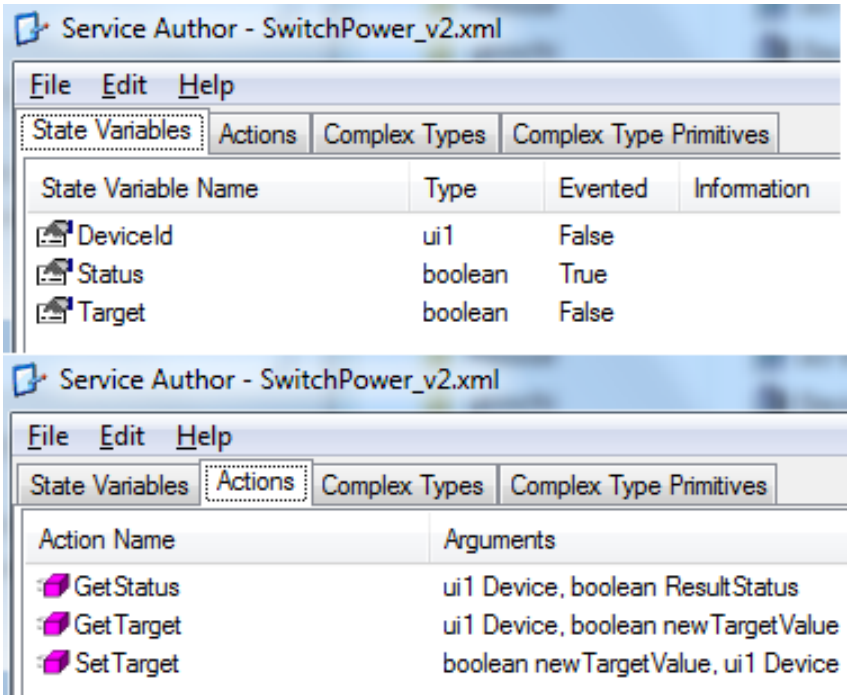
Figura 17 - Telas do Service Author para o serviço de dimerização



Fonte: autor

O serviço de dimerização tem a variável de estado, *LoadLevelStatus*, que pode variar de 0 a 100, e a variável que identifica a lâmpada que a função está acionando, *DeviceId*. A variável *LoadLevelTarget* é a entrada da função, e a *MinLevel* é um limitador de nível.

Figura 18 - Telas do Service Author para o serviço de liga/desliga

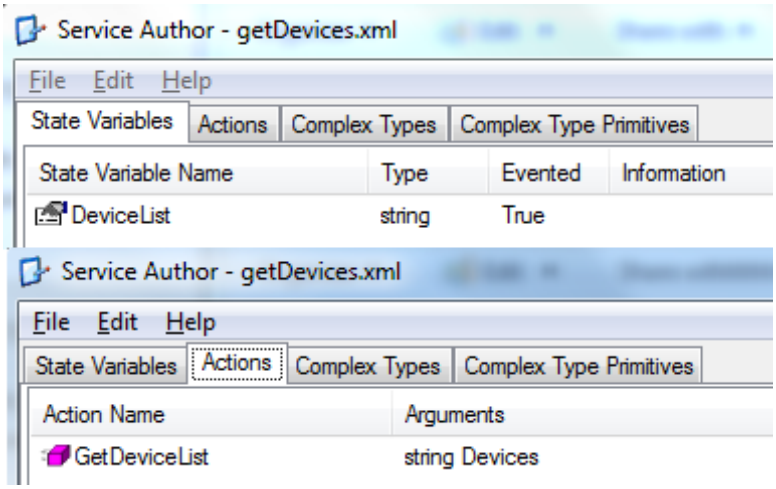


Fonte: autor

O serviço de liga/desliga tem a variável de estado booleana, *Status*, para indicar se a lâmpada está acesa ou apagada. Este serviço também tem a variável de identificação da lâmpada, *DeviceId*, e uma variável de entrada das funções, *Target*.

Dessa forma, a mesma definição de um serviço pode ser utilizada por vários dispositivos, passando o identificador do dispositivo a ser controlado. Porém, o cliente UPnP deve saber qual dispositivo será acionado. Para que o cliente UPnP saiba quais os dispositivos estão disponíveis e seus tipos, o sistema tem serviços básicos de descrição de seus dispositivos. A Figura 19 mostra as telas do *Service Author* para esse serviço.

Figura 19 - Serviço do subsistema de iluminação



Fonte: autor

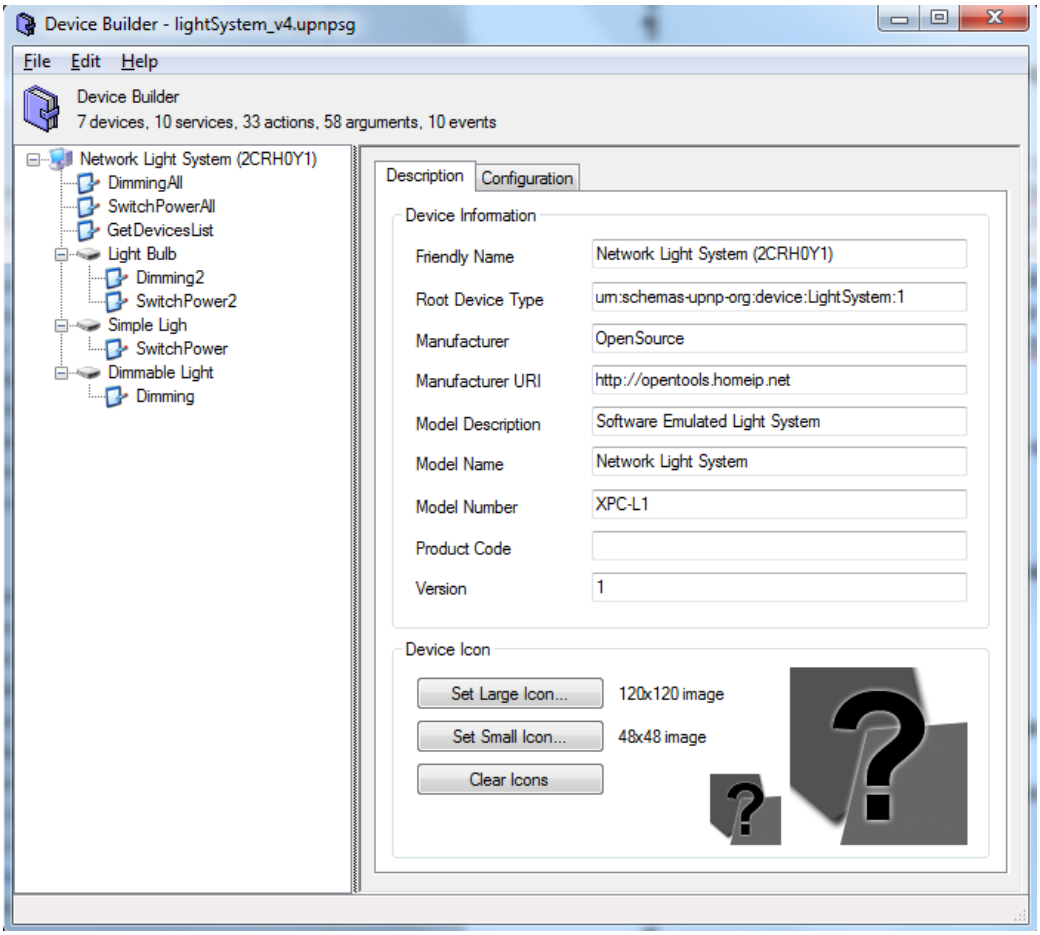
O sistema de iluminação tem um serviço, chamado *getDevices*, que retorna a descrição e mapeamento de suas lâmpadas, quais são lâmpadas simples e quais são dimerizáveis e seus respectivos identificadores, e a lista de dispositivos do sistema, *DeviceList*. Os sistemas também podem ter serviços que retornam metadados sobre seus dispositivos, por exemplo, sua localização física dentro da residência entre outras informações que possam ajudar a interface com o usuário.

Subsistemas também podem ter serviços de controle geral do subsistema e de todos os dispositivos simultaneamente, por exemplo, o subsistema de iluminação pode ter um serviço para apagar todas as luzes com um único comando. Esses serviços do sistema podem implementar funcionalidades variadas, desde de administração do subsistema, até funcionalidades de inteligência artificial, por

exemplo, um serviço para ativar um cenário de simulação de presença de pessoas na residência enquanto os moradores estiverem viajando, no qual as lâmpadas são ligadas e desligadas automaticamente seguindo um padrão programado.

Com os serviços criados pode-se criar a descrição do sistema. Seguindo o UPnP estendido, o sistema de iluminação é o sistema raiz com os seus serviços. Nesse sistema, os tipos de dispositivos são lâmpadas simples e lâmpadas dimerizáveis, e cada tipo tem seus respectivos serviços. A Figura 20 mostra a tela da descrição do sistema de iluminação no *Device Builder*.

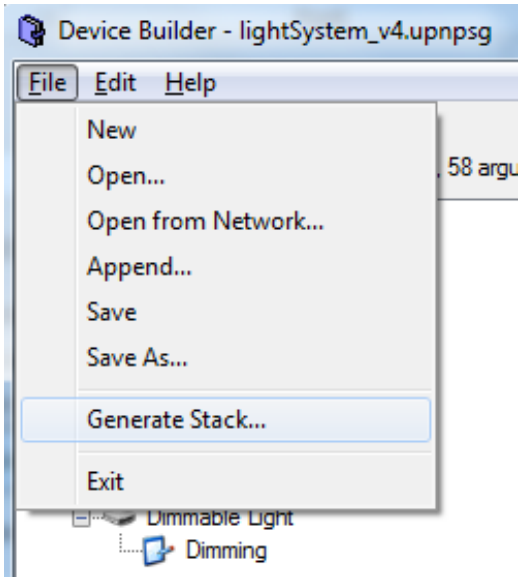
Figura 20 - Descrição do sistema de iluminação



Fonte: autor

Depois de criar a descrição do subsistema com o *Device Builder,* pode-se gerar a estrutura de código do sistema UPnP através dessa ferramenta. A Figura 21 apresenta o menu para gerar essa estrutura de código do subsistema.

Figura 21 - Geração da estrutura de código UPnP

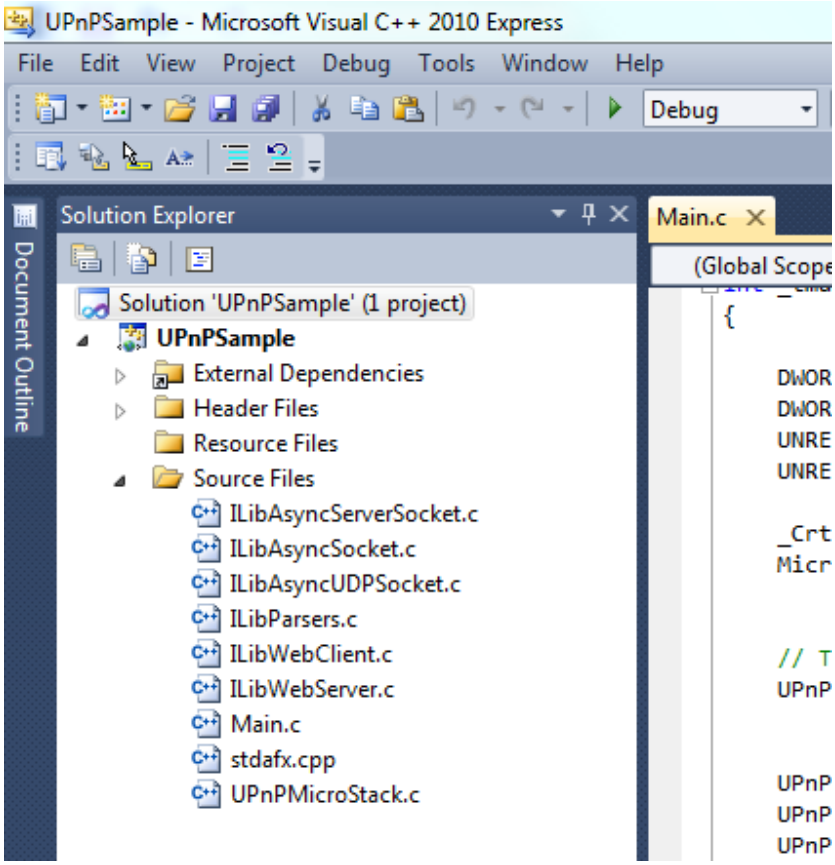


Fonte: autor

5.3.2 Código C do subsistema de iluminação

Ao gerar a estrutura de código com o *Device Builder,* a ferramenta cria na pasta escolhida todos os arquivos necessários para implementar o sistema, as bibliotecas do protocolo UPnP e o projeto do Visual Studio. O modelo de código com as funções referentes aos serviços é criado no arquivo ‘Main.c’. Para editar o código do subsistema de iluminação e compilá-lo, foi utilizado o Microsoft Visual Studio Express. A Figura 22 mostra a estrutura do projeto no Visual Studio.

Figura 22 - Projeto do Visual Studio

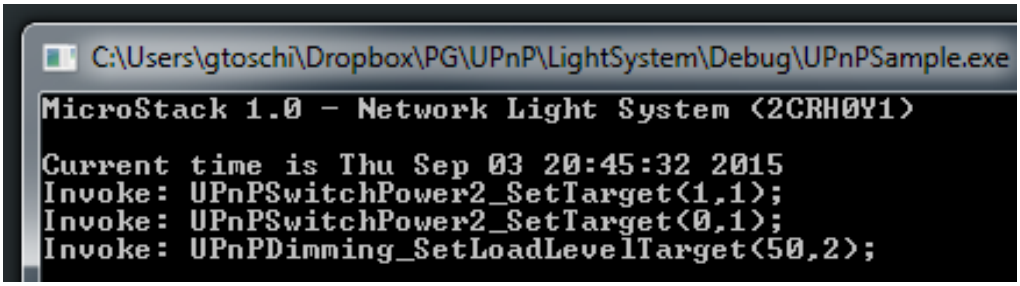


Fonte: autor

No arquivo Main.c encontram-se as funções referentes aos serviços criados e as variáveis internas de controle dos dispositivos do sistema de iluminação. O código completo dessas funções pode ser encontrado no Apêndice A.

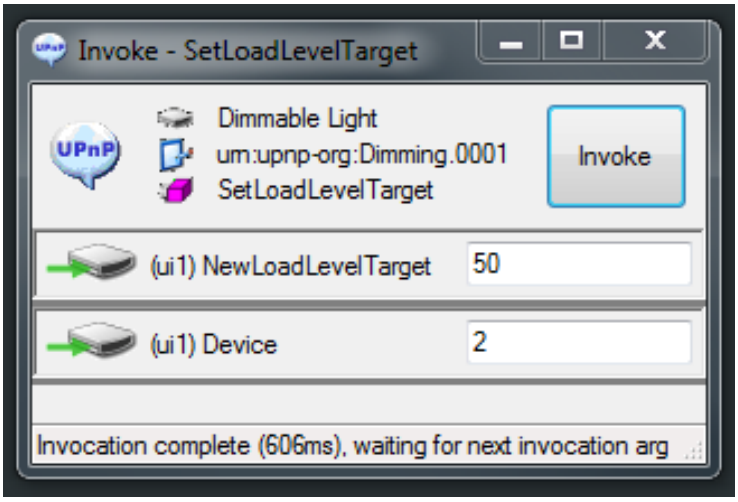
As funções dos serviços UPnP atualizam essas variáveis de controle, imprimem a ação realizada na janela da instância UPnP do sistema e escreve todas as ações em um arquivo de log unificado. Esse dispositivo simulado não tem uma interface gráfica, ou seja, a simulação do dispositivo virtual é uma janela de linha de comandos, apresentada na Figura 23. A Figura 24 apresenta a interface do cliente UPnP da ferramenta para um serviço de uma lâmpada do sistema de iluminação. Elas apresentam a interface do cliente UPnP invocando uma ação e o resultado da ação na janela do subsistema.

Figura 23 - Interface do sistema de iluminação



Fonte: autor

Figura 24 - Ação de uma lâmpada do sistema de iluminação



Fonte: autor

5.4 APRESENTAÇÃO DAS SIMULAÇÕES REALIZADAS

As simulações foram realizadas a partir dessa estrutura de dispositivos do subsistema virtual. Com o cliente UPnP pode-se utilizar os serviços do subsistema e seus dispositivos, e ao utilizar os serviços do subsistema de iluminação virtual criado, este não atua sobre dispositivos reais, mas grava suas ações em um arquivo de *log*, juntamente com a data e hora da ação realizada, para que depois possa ser realizada uma análise.

Uma análise qualitativa foi realizada com foco na usabilidade do modelo estendido em relação ao modelo original. Com foco nessa análise, foram realizadas simulações de controle individual de dispositivos, além do controle conjunto de dispositivos através de serviços do subsistema. O cliente UPnP da ferramenta é simples, com uma interface concisa, porém permite o acesso direto a todos os

serviços do subsistema de iluminação e de suas lâmpadas, além de possibilitar que o usuário escolha o valor de todos os parâmetros dos serviços disponíveis manualmente.

Um parâmetro que está sempre presente nos serviços dos dispositivos é o identificador do dispositivo. Como a interface do serviço é igual para todos os dispositivos, a solução encontrada foi adicionar o parâmetro de identificação do dispositivo que está sendo acionado. Dessa forma, o subsistema sabe qual dispositivo acionar, isto é, a central do subsistema de iluminação, que expõem a interface UPnP, consegue identificar para qual dispositivo enviar o comando. Por exemplo, no subsistema de iluminação, a central que está conectada à rede IP da residência consegue controlar a lâmpada especificada, através da sub-rede *mesh* ZigBee, que conecta todos os dispositivos desse subsistema.

5.5 FECHAMENTO

Neste capítulo foram apresentados o caso de uso, a ferramenta de construção, a implementação do sistema e sua simulação. Em sequência os resultados das simulações serão apresentados e discutidos para comparar o padrão antes e depois da extensão proposta.

6. RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados das simulações realizadas com a extensão do modelo do UPnP apresentado no capítulo anterior.

6.1 ANÁLISE DOS RESULTADOS OBTIDOS

Para realizar as simulações foi utilizado o cliente UPnP simplificado da ferramenta *Developer Tools for UPnP*, e neste o usuário tem acesso direto a todas as funções e suas variáveis. Ao realizar as simulações pode-se perceber a maior complexidade necessária do cliente UPnP, pois ele deve identificar o dispositivo alvo, através de seu identificador, em todas as funções.

Então antes de conseguir utilizar as funções dos dispositivos do subsistema de iluminação, o cliente UPnP deve obter a descrição do subsistema que mapeia os dispositivos, identificadores e seus tipos respectivos. A partir desse mapeamento o cliente sabe qual interface chamar e qual identificador corresponde de cada dispositivo, e assim, quando o usuário pretende controlar um dispositivo específico, o cliente UPnP sabe qual identificador utilizar como entrada na função utilizada.

No cliente UPnP simples utilizado, o campo de identificação do dispositivo é um campo aberto que o usuário deve preencher manualmente. Porém, em um cliente UPnP utilizado pelo usuário final, a interface gráfica pode ser mais elaborada e atrativa. Pode-se, por exemplo, utilizar um mapa digital da residência no qual o usuário pode navegar para achar o dispositivo desejado e assim entrar em sua interface de controle. Quando o usuário identificar graficamente o dispositivo, o cliente UPnP já sabe como identificá-lo e qual interface utilizar para controlá-lo.

As simulações foram realizadas seguindo esse processo, utilizando esse cliente UPnP simplificado manualmente para utilizar iterativamente as funções de controle dos dispositivos e do sistema de iluminação construído. Essas simulações geraram *logs* que foram analisados para identificar o comportamento do novo padrão.

Esse novo padrão estendido, apesar de gerar uma complexidade extra no cliente UPnP, permite uma maior flexibilidade ao sistema, e aliado à arquitetura

proposta, permite a modularização do sistema, dividindo-o em subsistemas. Porém as simulações foram realizadas com o subsistema de iluminação construído.

Em um sistema de iluminação que utilize o padrão UPnP sem a extensão, cada dispositivo teria uma interface UPnP dedicada embarcada, isto é, cada lâmpada deveria ser capaz de processar todo esse protocolo em seu próprio hardware. Utilizando o padrão estendido, cria-se um subsistema com a interface UPnP centralizada, que utiliza uma tecnologia de comunicação especializada para o tipo de dispositivo, isto é, a central do subsistema de iluminação está conectada na rede IP doméstica para expor a interface de controle UPnP, e pode utilizar uma rede *mesh* ZigBee, por exemplo, para se comunicar com as lâmpadas.

Mesmo com essa diferença na arquitetura, a partir das simulações pôde-se observar que o controle individual dos dispositivos não foi impactado com a modularização em subsistema. O controle de um dispositivo UPnP e de um dispositivo em um subsistema foi igual. Para o usuário final o controle dos dispositivos não foi afetado, e ele ainda tem os mesmos controles sobre cada dispositivo. Em um sistema com o padrão sem a extensão, ter-se-ia interfaces UPnP iguais, uma para cada dispositivo, expostas na rede doméstica, já com o padrão estendido, tem-se uma interface igual a de um dispositivo, mas com um parâmetro a mais para a identificação do dispositivo a ser controlado, o identificador do dispositivo, *DeviceId*.

Também foi observado que se possibilitou a criação de funções do subsistema para o controle em grupo dos dispositivos. Esse controle em grupo, além de aumentar o valor agregado do subsistema, também descongestiona a rede doméstica, pois a central utiliza a rede interna do subsistema para fazer o controle e vigilância dos dispositivos. Em um sistema sem a extensão, esse controle em grupo também poderia ser construído, porém de uma forma distinta. Um cliente UPnP mais complexo poderia disponibilizar para o usuário uma interface de controle em grupo, por exemplo. Dessa forma, quando o usuário utilizaria um controle em grupo, o cliente UPnP dispararia sequencialmente os comandos de controle para as interfaces UPnP de cada dispositivo.

Porém, essa solução não depende da interface do cliente UPnP que o usuário escolher, e também funcionaria com um sistema que utiliza o padrão estendido. Em comparação com as funções em grupo do padrão estendido, essa solução pode

congestionar a rede IP doméstica por alguns segundos, dependendo do número de dispositivos acionados nesse controle em grupo. Essa sobrecarga da rede não ocorre quando se utiliza a função de grupo do subsistema com o padrão estendido, no qual o cliente UPnP envia apenas um comando para o subsistema, e este, utilizando sua sub-rede, controla os vários dispositivos do grupo de forma otimizada.

Ao se separar a comunicação externa e interna ao subsistema, isto é, a rede IP doméstica e a sub-rede especializada do subsistema, pode-se observar um menor volume de dados na transferência da descrição do subsistema para o cliente UPnP quando o padrão estendido é utilizado. Um exemplo de comparação é em um subsistema que tem 200 dispositivos de dois tipos distintos, no qual os dados transferidos, toda vez que o cliente precisar se atualizar, terá apenas duas descrições, ao invés de duzentas, isto é, uma por tipo ao invés de uma por dispositivo do subsistema.

Apesar dessa modularização em subsistemas, as vantagens do padrão UPnP são mantidas, como a padronização das interfaces dos dispositivos, o que proporciona a facilidade de integração entre subsistemas distintos através da camada utilizando o UPnP estendido proposto. A entidade que criou e mantém o padrão UPnP teve como objetivo a integração transparente de dispositivos compatíveis, e a extensão proposta manteve essa característica e a utilizou para criar a integração transparente entre subsistemas de automação. O sistema de automação residencial é assim composto de módulos, subsistemas de automação, integrados por essa camada de comunicação que utiliza a extensão do padrão UPnP, e dessa forma cria-se a característica *plug and play* de subsistemas de automação em uma residência.

Essa característica flexibiliza e facilita a compra e manutenção do sistema de automação residencial. O usuário final poderá, então, escolher módulos distintos para automatizar sua residência, satisfazendo melhor as suas necessidades, e incorporá-los gradualmente, isto é, o usuário poderá comprar subsistemas de automação de forma progressiva e adicioná-los ou substituí-los, sem que seja necessário a fidelização a uma marca específica. Essa flexibilização proporcionada tem potencial para fomentar a maior adoção desses sistemas nas residências, aumentando a inserção dessa tecnologia nas vidas das pessoas.

6.2 COMPARATIVOS COM AS SOLUÇÕES JÁ EXISTENTES

Uma análise qualitativa da extensão proposta pode ser realizada através da comparação entre o padrão UPnP estendido com outros padrões encontrados no mercado, utilizando-se a Análise SWOT ou FFOA (Forças/*Strengths*, Fraquezas/*Weaknesses*, Oportunidades/*Opportunities* e Ameaças/*Threats*). O objetivo desta análise é efetuar a síntese das análises internas e externas, identificando os pontos positivos e indicando quais pontos devem melhorar.

O Quadro 3 e o Quadro 4 apresentam a analise SWOT do padrão estendido e do padrão atual.

Quadro 3 - Análise SWOT do UPnP estendido

	Facilita	**Dificulta**
Ambiente interno	• Controle individual ou em grupo dos dispositivos do subsistema. • Modularização do sistema de automação residencial em subsistemas. • Subsistemas podem ser utilizados independentes ou em conjunto. • Possibilidade de inserir meta-dados sobre os dispositivos. • Padronização da interface dos dispositivos.	• Aumento da complexidade do cliente UPnP. • Configuração dos dispositivos da residência no sistema.
Ambiente externo	• Penetração do UPnP nas residências já existentes, presentes em quase todos os roteadores residenciais atuais. • Mercado em crescimento. • Facilidade de instalação. • Utilização da infraestrutura de rede doméstica já existente.	• Adoção do UPnP como camada de integração de subsistemas.

Fonte: autor

Quadro 4 - Análise SWOT padrões atuais

	Facilita	**Dificulta**
Ambiente interno	• Diversidade de dispositivos, empresas e *startups*. • Novos protocolos estão sempre sendo criados.	• Protocolos de comunicação que devem ser utilizados verticalmente. • Utilização de tecnologias distintas e incompatíveis. • Falta de integração das soluções existentes.
Ambiente externo	• Diversidade de soluções disponíveis. • Sistemas autônomos de automação. • Mercado em crescimento.	• Não existe um padrão de integração adotado universalmente. • Tecnologias proprietárias que obrigam a fidelização do usuário.

Fonte: autor

6.3 DISCUSSÕES

Muitas iniciativas vêm sendo conduzidas para melhorar padrões e criar métodos de integrá-los. Porém essas melhorias na automação residencial não são garantia de aumento na adoção desta de forma abrangente.

A futura maturidade do mercado de automação residencial pode vir através da padronização do mercado, com a utilização e adoção de um único padrão em todas as residências, mas também através da diversificação do mercado, com a consolidação de diversos padrões que conseguem interagir entre si. Mas ambas as situações podem demorar a acontecer, e mesmo com a estabilização do mercado, a

automação residencial pode não se tornar ubíqua e onipresente, adotada em todas as residências.

A tendência que prevalecerá é aquela que permite novas tecnologias serem adicionadas e integradas de modo transparente, fazendo com que novas soluções se tornem compatíveis com os sistemas de automação residencial existentes. Outro fator que deve influenciar no sucesso do padrão é sua integração com outros sistemas, por exemplo, a integração da residência com os sistemas de infraestrutura da cidade, em uma *smart-city*, no qual as residências estarão integradas com a cidade.

Essas novas conectividades são baseadas na conexão da residência à Internet, e assim surgem novos riscos de segurança da residência, como abertura do controle da residência à internet ou a convidados, que devem ser mitigados, para que vulnerabilidades não sejam adicionadas junto com novas funcionalidades.

7. CONCLUSÕES

Este trabalho apresentou e discutiu uma proposta de extensão do padrão UPnP, focada na automação residencial, com uma arquitetura dividida em duas camadas, uma abstrata de integração e outra especializada para o controle dos dispositivos. A extensão e a arquitetura foram propostas em conjunto, pois são complementares e funcionam em conjunto para flexibilizar e modularizar o sistema de automação residencial proposto.

Esse sistema é modularizado em subsistemas autônomos e especializados, que estão integrados pela camada do UPnP. Dessa forma esse sistema pode utilizar a infraestrutura doméstica já presente, facilitando a instalação tanto em residências antigas como novas, já construídas ou em projeto.

Essa modularização em subsistemas permite, por exemplo, a integração de dispositivos simples, como uma lâmpada, à uma infraestrutura complexa que utiliza protocolos de alto nível, como o UPnP, utilizando a rede Wi-Fi. Dessa forma, dispositivos com restrições de consumo de energia, processamento, etc. podem ser integrados no sistema de automação residencial sem a necessidade de processar um protocolo de alto nível. Os subsistemas podem utilizar protocolos de comunicação interna mais adequados para o tipo de dispositivo, como o ZigBee, que exigem menor processamento e apresentam baixo consumo de energia, além de poder, por exemplo, estender seu alcance através da utilização do modo *mesh*.

Além disso, o custo de produção dos dispositivos pode diminuir, pois cada um deles pode ser construído com a menor complexidade necessária possível, para utilizar a tecnologia de comunicação especializada do subsistema. Logo, seus requisitos de hardware podem ser menores e os dispositivos podem ser montados com componentes mais simples e de menor custo. Entretanto, o subsistema necessita uma central de controle que conecta os dispositivos à infraestrutura UPnP através do cliente UPnP estendido, e dessa forma, os fabricantes ficam encarregadas de desenvolver esse controle interno dos subsistemas e implementar a interface UPnP compatível com o padrão, e o usuário precisa apenas se preocupar em escolher o subsistema com as funcionalidades que melhor atendam às suas necessidades.

Essas melhorias propostas foram pesquisadas com o objetivo de aumentar a adoção e popularização da automação residencial pela população em geral. Para isso tentou-se solucionar os problemas que os usuários encontram em sistemas de automação residencial e facilitar a compra e manutenção desses sistemas para tentar aumentar sua adoção. Porém, este objetivo só poderá ser comprovado de fato após alguns anos, caso a proposta feita neste trabalho seja adotada.

A proposta apresentada também pode gerar dificuldades para a sua adoção na automação residencial. A utilização de descrições dos tipos de dispositivos e suas interfaces padrões faz com que o cliente UPnP necessite coletar a descrição do sistema, com o mapeamento dos dispositivos e seus respectivos tipos, e a meta-informação, o que aumenta a complexidade dos clientes UPnP. Anteriormente, esses clientes poderiam ler a descrição de todos os dispositivos do sistema e expô-los diretamente aos usuários, mas, com a extensão proposta, os clientes devem coletar esse mapeamento dos dispositivos e seus tipos, para então poder expor e utilizar a interface correta para cada dispositivo no sistema de automação. Porém, com a evolução do hardware e dos *smartphones*, esse aumento da complexidade tende a não ser um problema para os clientes UPnP embarcados.

7.1 PERSPECTIVAS DE CONTINUIDADE DO TRABALHO

Os trabalhos futuros relacionados ao presente trabalho podem ser divididos em duas vertentes: uma relacionada com a continuidade das pesquisas, e outra com a produção de sistemas de automação residencial baseados no padrão proposto. As pesquisas futuras poderiam ser conduzidas no sentido de possibilitar a obtenção de respostas à perguntas que ainda estão em aberto, e também possibilitar análises quantitativas e de pesquisa de mercado com o consumidor.

Já a produção de sistemas de automação residencial pode ser alcançada com a criação de uma nova empresa, por exemplo, fomentada por uma incubadora da USP. A empresa poderia desenvolver um módulo de interface de comunicação e controle que utiliza a extensão do UPnP para conectar os subsistemas. Dessa forma, poder-se-ia desenvolver fornecedores parceiros que se interessassem em integrar em seus sistemas essa interface UPnP, e estes poderiam ser vendidos como um subsistema de um sistema de automação.

Outra alternativa seria a pesquisa associada à criação da empresa. Dessa forma, um projeto de pesquisa poderia ser conduzido pelos pesquisadores associados à empresa criada e fomentada por um fundo de pesquisa.

7.2 FECHAMENTO

A pesquisa realizada neste trabalho foi árdua e longa, porém motivadora para aplicar os resultados teóricos e experimentais obtidos na criação de sistemas integrados de automação residencial, e também para continuar os estudos e pesquisas neste tema atualmente em destaque e prestigiado. Trouxe, também, um grande aprendizado sobre a pesquisa e escrita formal e científica, sobre a carreira acadêmica e sobre o comprometimento e dedicação necessária para se concluir uma pesquisa.

8. REFERÊNCIAS

AL-MUHTADI, J.; ANAND, M.; MICKUNAS, M. D.; CAMPBELL, R. Secure smart homes using Jini and UIUC SESAME. In: Computer Security Applications, 2000. ACSAC '00, 16th Annual Conference, 16, 2000, New Orleans, LA. **Anais ...** New Orleans, LA: IEEE, 2000. p.77–85.

ÅSTRÖM, K. J. Process control: Past, present and future. **IEEE Control System Magazine**, v. 5, p. 3–10, 1985.

ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. **Computer Networks**, v. 54, n. 15, p. 2787–2805, 2010. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128610001568>. Acesso em: 11/2/2013.

BALTA-OZKAN, N.; DAVIDSON, R.; BICKET, M.; WHITMARSH, L. Social barriers to the adoption of smart homes. **Energy Policy**, v. 63, p. 363–374, 2013. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0301421513008471>. Acesso em: 7/11/2015.

BOLZANI, C. A. M. Desmistificando a domótica. **Escola Politécnica da Universidade de São Paulo**, 2007. Disponível em: <http://www.cefetsp.br/edu/prp/sinergia/complemento/sinergia_2007_n1/pdf_s/segmentos/artigo_02_v8_n1.pdf>. Acesso em: 29/3/2013.

BOLZANI, C. A. M. **Residências Inteligentes: Domótica, Redes Domésticas e Automação Residencial.** 1º ed. São Paulo, Brasil: Editora Livraria da Física, 2004.

BOULOS, M. N.; WHEELER, S.; TAVARES, C.; JONES, R. How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX. **Biomedical Engineering Online**, v. 10, p. 24, 2011. Disponível em: <http://link.springer.com/article/10.1186%2F1475-925X-10-24>. Acesso em: 27/8/2015.

BUYS, L.; BARNETT, K. R.; MILLER, E.; BAILEY, C. Smart housing and social sustainability: Learning from the residents of Queensland's Research House. **Australian Journal of Emerging Technologies and Society**, v. 3, n. 1, p. 43–57, 2005. Disponível em: <http://eprints.qut.edu.au/6065/>. Acesso em: 10/3/2015.

CHAN, M.; ESTÈVE, D.; ESCRIBA, C.; CAMPO, E. A review of smart homes—Present state and future challenges. **Computer Methods and Programs in Biomedicine**, v. 91, n. 1, p. 55–81, 2008.

CHEN, M.; WAN, J.; GONZALEZ, S.; LIAO, X.; LEUNG, V. C. M. A Survey of Recent Developments in Home M2M Networks. **IEEE Communications Surveys and Tutorials**, v. 16, n. 1, p. 98–114, 2014.

CHESHIRE, A.; ABOBA, B. Dynamic configuration of IPv4 link-local addresses, RFC 3927. **IETF**, 2005. Disponível em: < https://tools.ietf.org/html/rfc3927 >. Acesso em:7/6/2014.

COHEN, J.; AGGARWAL, S.; GOLAND, Y. General event notification architecture base: Client to arbiter. **IETF**, 1999. Disponível em: < https://tools.ietf.org/html/rfc3927 >. Acesso em:25/8/2014.

CONSIDINE, T.; BOGEN, C. OASIS Open Building Information Exchange (oBIX) TC. **OASIS**, 2014. Disponível em: <https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=obix>. Acesso em: 21/10/2014.

COOK, D. J.; AUGUSTO, J. C.; JAKKULA, V. R. Ambient intelligence: Technologies, applications, and opportunities. **Pervasive and Mobile Computing**, v. 5, n. 4, p. 277–298, 2009.

COOK, D. J.; DAS, S. K. How smart are our environments? An updated look at the state of the art. **Pervasive and Mobile Computing**, v. 3, p. 53–73, 2007.

DAHM, M.; MATHUR, A. Automation in the food processing industry: distributed control systems. **Food Control**, v. 1, p. 32–35, 1990.

DEMIRIS, G.; HENSEL, B. “Smart Homes” for Patients at the End of Life. **Journal of Housing For the Elderly**, v. 23, n. 1-2, p. 106–115, 2009.

DEN HARTOG, F.; TIJMES, M.; CRAMER, E.; NOORBERGEN, A. **Enabling new e-health business models by converging IP-based and non-IP-based home networks**. New York: IEEE, 2013.

Developer Tools - Open Software Projects. **OPENTOOL,** 2007. Disponível em: <http://opentools.homeip.net/dev-tools-for-upnp>. Acesso em: 22/12/2015.

DLNA. DLNA Overview and Vision Whitepaper 2007. **DLNA**, 2007. Disponível em: <https://wikileaks.org/sony/docs/05/docs/DLNA/DLNA_white_paper.pdf>. Acesso em: 22/12/2015.

ECHELON. LonWorks Protocol. **ECHELON**, 2012 Disponível em: <http://www.echelon.com/technology/lonworks/lonworks-protocol.htm>. Acesso em: 24/5/2015.

EUROPEAN COMMISSION. European Commission : CORDIS : Projects and Results : Amigo Ambient Intelligence for the networked home environment. **CORDIS**, 2004. Disponível em: <http://cordis.europa.eu/project/rcn/71920_en.html>. Acesso em: 21/10/2014.

FAGNANT, D. J.; KOCKELMAN, K. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. **Transportation Research Part A: Policy and Practice**, v. 77, p. 167–181, 2015.

FEHRENBACHER, K. Report: Google getting back into the energy data game? — Tech News and Analysis. **Gigaom**, 2013. Disponível em: <http://gigaom.com/2013/12/16/report-google-getting-back-into-the-energy-data-game/>. Acesso em: 15/6/2014.

FERREIRA, H. G. C.; CANEDO, E. D.; DE SOUSA JUNIOR, R. T. IoT Architecture to Enable Intercommunication Through REST API and UPnP Using IP, ZigBee and Arduino. **2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (wimob)**, Lyon: IEEE, outubro de 2013. p. 53–60.

FERRUS, R.; SALLENT, O.; PEREZ-ROMERO, J.; AGUSTI, R. A Solution Framework to Provide Management Services for Wireless Communications in the Digital Home. **IEEE Communications Magazine**, v. 50, n. 11, p. 132–141, 2012.

FISHER, D. BACnet and LonWorks: a white paper. **AIRAH Journal**, v. 54, n. 2, 2000.

GAMA, K.; TOUSEAU, L.; DONSEZ, D. Combining heterogeneous service technologies for building an Internet of Things middleware. **Computer Communications**, v. 35, n. 4, p. 405–417, 2012.

GENG, Y.; CASSANDRAS, C. New “Smart Parking” system based on resource allocation and reservations. **IEEE Transactions on Intelligent Transportation Systems**, v. 14, p. 1129–1139, 2013.

GIL, A. C. **Como Elaborar Projetos de Pesquisa**. 4º ed. São Paulo, Brasil: Atlas, 2002.

GILL, K.; YAO, F.; YANG, S.-H. Transparent Heterogeneous Networks for Remote Control of Home Environments. In: IEEE International Conference on Networking, Sensing and Control (ICNSC 2008), 2008, 5, Sanya, China. **Anais...** Sanya, China: IEEE, 2008. p.1419–1424.

GOLAND, Y.; CAI, T.; LEACH, P.; GU, Y.; ALBRIGHT, S. Simple service discovery protocol/1.0, operating without an Arbiter. **IETF**, 1999. Disponível em: <https://tools.ietf.org/html/draft-cai-ssdp-v1-03>. Acesso em: 7/6/2014.

GOOGLE PRESS. Google to Acquire Nest – Investor Relations – Google. **Investor**, 2014. Disponível em: <https://investor.google.com/releases/2014/0113.html>. Acesso em: 15/6/2014.

GUDGIN, M.; HADLEY, M.; MENDELSOHN, N.; MOREAU, J.; NIELSEN, H. Simple object access protocol (SOAP) 1.2, W3C Recommendation. **W3C**, 2003.

GUGLIELMO, C. CES LIVE: Cisco's Chambers Says Internet of Everything, $19 Trillion Opportunity, Is Next Big Thing. **FORBES**, 2014. Disponível em: <http://www.forbes.com/sites/connieguglielmo/2014/01/07/ces-live-cisco-ceo-chambers-to-deliver-keynote/>. Acesso em: 15/6/2014.

GUINARD, D.; TRIFA, V.; KARNOUSKOS, S.; SPIESS, P.; SAVIO, D. Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services. **IEEE Transactions on Services Computing**, v. 3, n. 3, p. 223–235, 2010.

HYDRA, P. The Hydra objective: Middleware for networked embedded system. Disponível em: <http://www.hydramiddleware.eu/>. .

IEC, I. ISO/IEC 25000:2005 - Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE. **ISO**, 2005. Disponível em: <http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=35683>. Acesso em: 31/8/2013.

INTEL DEVELOPER TOOL. **INTEL**, 2011. Disponível em: <https://software.intel.com/en-us/blogs/2011/03/07/developer-tools-for-upnp-tech-now-with-androidjava-support?language=es&wapkw=upnp+tools>. Acesso em 04 abril 2014.

INTEL UPNP EXAMPLE. **INTEL**, 2011. Disponível em: <https://noggin.intel.com/content/upnp-design-by-example-ch-13> e <https://noggin.intel.com/content/upnp-design-by-example-ch-12>. Acesso em 04 abril 2014.

INTEL UPNP TOOLS. **INTEL**, 2011. Disponível em: <https://software.intel.com/en-us/node/146715?wapkw=upnp+tools>. Acesso em 04 abril 2014.

INTILLE, S. S. Designing a home of the future. **IEEE pervasive computing**, v. 1, p. 76–82, 2002.

INTILLE, S. S. The goal: smart people, not smart homes. In: Proceedings of ICOST2006, 4, 2006, Amsterdam. **Anais...** Amsterdam: IOS Press, 2006. p.3–6.

JAI, B.; OGG, M.; RICCIARDI, A. Effortless software interoperability with Jini connection technology. **Bell Labs Technical Journal**, v. 5, n. 2, p. 88–101, 2000.

JANSE, M.; VINK, P.; GEORGANTAS, N. Amigo Architecture: Service Oriented Architecture for Intelligent Future In-Home Networks. In: M. Mühlhäuser; A. Ferscha; E. Aitenbichler (Orgs.); **Constructing Ambient Intelligence**, Communications in Computer and Information Science. p.371–378, 2008. Springer Berlin Heidelberg. Disponível em: <http:\/\/link.springer.com\/chapter\/10.1007\/978-3-540-85379-4_41>. Acesso em: 21/10/2014.

KAILAS, A.; CECCHI, V.; MUKHERJEE, A. A Survey of Communications and Networking Technologies for Energy Management in Buildings and Home Automation. **Journal of Computer Networks and Communications**, v. 2012, p. e932181, 2012.

KALAIVANI, T.; ALLIRANI, A.; PRIYA, P. A survey on ZigBee based wireless sensor networks in agriculture. In: 2011 3rd International Conference on Trendz in Information Sciences and Computing (TISC), 3, 2011, Chennai, India. **Anais...** Chennai, India: IEEE, 2011. p.85–89.

KIM, D.-S.; LEE, J.-M.; KWON, W.-H.; YUH, I. K. Design and implementation of home network systems using UPnP middleware for networked appliances. **IEEE Transactions on Consumer Electronics**, v. 48, n. 4, p. 963–972, 2002.

KIM, E.-J.; YOUM, S. Machine-to-machine platform architecture for horizontal service integration. **Eurasip Journal on Wireless Communications and Networking**, p. 79, 2013.

KIM, J.; LEE, J.; YUN, J. M2M Service Platforms: Survey, Issues, and Enabling Technologies. **IEEE Communications Surveys Tutorials**, v. 16, n. 1, p. 61–76, 2014.

KING, N. Smart home: a definition. **Scie-socialcare:** Milton Keynes, 2003. Disponível em: <http://www.scie-socialcareonline.org.uk/smart-home-a-definition/r/a11G00000017xLiIAI>. Acesso em: 14/7/2015.

KNX. KNX Handbook for Home and Building Control. **KNX**, 2006. Disponível em: <http://www.knx.org/fileadmin/downloads/09%20-%20Various/04%20-%20KNX%20Handbook%205th%20edition/English/SILVERBIBLE-Book_EN_content.pdf>. Acesso em: 22/12/2015.

KORENCIAK, D. Application of LONWORKS technology in intelligent buildings. In: ELEKTRO, 3, 2012, Rajeck Teplice, Slovakia. **Anais ...** Rajeck Teplice, Slovakia: ELEKTRO, 2012. p.319–322.

KRONER, W. M. An intelligent and responsive architecture. **Automation in Construction**, v. 6, n. 5–6, p. 381–393, 1997.

KUIPERS, B.; BYUN, Y. T. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. **Robotics and autonomous systems**, v. 8, p. 47–63, 1991.

KUMAR, A.; MIHOVSKA, A.; KYRIAZAKOS, S.; PRASAD, R. Visible Light Communications (VLC) for Ambient Assisted Living. **Wireless Personal Communications**, v. 78, n. 3, p. 1699–1717, 2014.

KYSELYTSYA, Y.; WEINZIERL, T. Implementation of the KNX Standard. **Proceedings …** Vienna, Austria: KNX Scientific Conference, 2006.

LEINER, B. M.; CERF, V. G.; CLARK, D. D.; et al. A Brief History of the Internet. **SIGCOMM Computer Communication Review**, v. 39, n. 5, p. 22–31, 2009.

LERTLAKKHANAKUL, J.; CHOI, J. W.; KIM, M. Y. Building data model and simulation platform for spatial interaction management in smart home. **Automation in Construction**, v. 17, n. 8, p. 948–957, 2008.

LI, Q.; LIU, F.; ZHANG, G.; LIN, J. The Research of DLNA-based Digital Media Adapter. In: Third International Conference on Pervasive Computing and Applications, 3, 2008, Alexandria. **Anais...** Alexandria: IEEE, 2008. v. 1, p.467–472.

LIN, C.-T.; LIN, B.-S.; LIN, F.-C.; CHANG, C.-J. Brain Computer Interface-Based Smart Living Environmental Auto-Adjustment Control System in UPnP Home Networking. **IEEE Systems Journal**, v. 8, n. 2, p. 363–370, 2014.

LYYTINEN, K.; YOO, Y. Issues and Challenges in Ubiquitous Computing. **Communications of ACM**, v. 45, n. 12, 2002.

MANYIKA, J.; CHUI, M.; BUGHIN, J.; et al. Disruptive technologies: Advances that will transform life, business, and the global economy. **Chrysalix Enrgey Venture Capital**, 2013.

MCAFEE, A.; BRYNJOLFSSON, E. Big Data: The Management Revolution. **HBR**, 2012. Disponível em: <https://hbr.org/2012/10/big-data-the-management-revolution>. Acesso em: 27/8/2015.

MCQUILLAN, J. M.; RICHER, I.; ROSEN, E. C. The new routing algorithm for the ARPANET. **IEEE Transactions on Communication**, v. 28, p. 711–719, 1980.

MESHKOVA, E.; RIIHIJAERVI, J.; PETROVA, M.; MAEHOENEN, P. A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks. **Computer Networks**, v. 52, n. 11, p. 2097–2128, 2008.

MILLER, B. A.; NIXON, T.; TAI, C.; WOOD, M. D. Home networking with Universal Plug and Play. **IEEE Communications Magazine**, v. 39, n. 12, p. 104–109, 2001.

MIN, D.; LEE, J.; CHOI, E. The Active Embedded Ubiquitous Web Service Framework. In: D. Hutchison; S. Denazis; L. Lefevre; G. J. Minden (Orgs.); **Active and Programmable Networks**. v. 4388, p.212–217, 2009. Berlin: Springer-Verlag Berlin.

MIORANDI, D.; SICARI, S.; DE PELLEGRINI, F.; CHLAMTAC, I. Internet of things: Vision, applications and research challenges. **Ad Hoc Networks**, v. 10, n. 7, p. 1497–1516, 2012.

MIORI, V.; RUSSO, D. Domotic Evolution towards the IoT. In: 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 28, 2014, ICPCA. **Anais...** Victoria, Canada: IEEE, 2014. p.809–814.

MIORI, V.; RUSSO, D.; ALIBERTI, M. Domotic Technologies Incompatibility Becomes User Transparent. **Communications of the ACM**, v. 53, n. 1, p. 153–157, 2010.

MIORI, V.; RUSSO, D.; CONCORDIA, C. Meeting People's Needs in a Fully Interoperable Domotic Environment. **Sensors**, v. 12, n. 12, p. 6802–6824, 2012.

MIORI, V.; TARRINI, L.; MANCA, M.; TOLOMEI, G. An open standard solution for domotic interoperability. **IEEE Transaction on Consumer Electronics**, v. 52, n. 1, p. 97–103, 2006.

MITSUGI, J.; SATO, Y.; OZAWA, M.; SUZUKI, S. An integrated device and service discovery with UPnP and ONS to facilitate the composition of smart home applications. In: 2014 IEEE World Forum on Internet of Things (WF-IoT), 2014, Seoul, Korea. **Anais...** Seoul, Korea: IEEE, 2014. p.400–404.

MITSUGI, J.; YONEMURA, S.; YOKOISHI, T. Reliable and Swift Device Discovery in Consolidated IP and Zig Bee Home Networks. **IEEE Transactions on Communications**, v. E96B, n. 7, p. 1837–1844, 2013.

NEUGSCHWANDTNER, M.; NEUGSCHWANDTNER, G.; KASTNER, W. Web Services in Building Automation: Mapping KNX to oBIX. In: 2007 5th IEEE International Conference on Industrial Informatics, 5, 2007, Vienna. **Anais ...** Vienna, Austria: IEEE, 2007. v. 1, p.87–92.

NGO, L. Service-oriented architecture for home networks. In: Seminar on Internet working, 1, 2007, Helsinki. **Anais ...** Helsinki: TKK, 2007.

OBIX. oBIX. **OBIX**, 2011. Disponível em: <http://www.obix.org>. Acesso em: 21/10/2014.

ORACLE. Oracle and Sun Microsystems | Strategic Acquisitions | Oracle. **ORACLE**, 2010. Disponível em: <https://www.oracle.com/sun/index.html?PHPSESSID=8fd76773d26875481e097a4a27c7a6a1>. Acesso em: 23/12/2015.

PAETZ, A.-G.; DÜTSCHKE, E.; FICHTNER, W. Smart Homes as a Means to Sustainable Energy Consumption: A Study of Consumer Perceptions. **Journal of Consumer Policy**, v. 35, n. 1, p. 23–41, 2011.

PALACIOS, J. M.; FERNANDEZ, M.; CORCHO, O.; MENDEZ, V. M.; PEREZ, J. M. G. Semantically enabling UPnP Networks of Multimedia Home Content. **IEEE Latin America Transactions**, v. 9, n. 4, p. 586–592, 2011.

RAYCHOUDHURY, V.; CAO, J.; KUMAR, M.; ZHANG, D. Middleware for pervasive computing: A survey. **Pervasive and Mobile Computing**, v. 9, n. 2, p. 177–200, 2013.

ROGERS, Y. Moving on from Weiser's Vision of Calm Computing: Engaging UbiComp Experiences. In: P. Dourish; A. Friday (Orgs.); **UbiComp 2006: Ubiquitous Computing**. v. 4206, p.404–421, 2006. Berlin, Heidelberg: Springer Berlin Heidelberg. Disponível em: <http://www.springerlink.com/index/10.1007/11853565_24>. Acesso em: 2/11/2013.

ROSE, B. Home networks: a standards perspective. **IEEE Communications Magazine**, v. 39, n. 12, p. 78–85, 2001.

RUTA, M. Auto-adapting homes to context: a semantic-based evolution of KNX standard protocol. **SISINFLAB**, 2012. Disponível em: <http://sisinflab.poliba.it/swottools/smartbuilding automation/pdf/ruta_et_al_KNX2013.pdf>. Acesso em: 22/09/2015.

RUTA, M.; SCIOSCIA, F.; DI SCIASCIO, E.; LOSETO, G. A semantic-based evolution of EIB Konnex protocol standard. In: 2011 IEEE International Conference on Mechatronics (ICM), 2011 a, Istanbul, Turquia. **Anais ...** Istanbul, Turquia: IEEE, 2011. p.773–778.

RUTA, M.; SCIOSCIA, F.; DI SCIASCIO, E.; LOSETO, G. Semantic-Based Enhancement of ISO/IEC 14543-3 EIB/KNX Standard for Building Automation. **IEEE Transactions on Industrial Informatics**, v. 7, n. 4, p. 731–739, 2011 b.

RUTA, M.; SCIOSCIA, F.; LOSETO, G.; DI SCIASCIO, E. KNX: A Worldwide Standard Protocol for Home and Building Automation: State of the Art and Perspectives. **SISINFLAB**, 2014. Disponível em: <http://sisinflab.poliba.it/publications/2014/RSLD14a/ruta_et_al_ ICTH2014.pdf>. Acesso em: 22/09/2015.

SANCHEZ, S.; PETROVA, M.; MAHONEN, P. UPnP service discovery for heterogeneous networks. **IEEE PIMRC**, 2006.

SARNOVSKÝ, M.; KOSTELNÍK, P.; BUTKA, P.; HRENO, J.; LACKOVÁ, D. First Demonstrator of HYDRA Middleware Architecture for Building Automation. In: Znalosti, 2008, Bratislava, Slovakia. **Anais ...** Bratislava, Slovakia: Research Gateway, 2008.

SATYANARAYANAN, M. Pervasive computing: vision and challenges. **IEEE Personal Communications**, v. 8, n. 4, p. 10–17, 2001.

TAN, S.; ZHANG, J.; SUN, F.; WANG, S. An Approach to Support the Interoperability of Intelligent Grouping and Resource Sharing (IGRS) and Universal Plug and Play (UPnP) in Home Network Environment. In: 2015 IEEE International Conference on Computational Intelligence Communication Technology (CICT), 2015, Ghaziabad. **Anais ...** Ghaziabad: IEEE, 2015.

TILLEY, A. Google Acquires Smart Thermostat Maker Nest For $3.2 Billion. FORBES, 2014. Disponível em: <http://www.forbes.com/sites/aarontilley/2014/01/13/google-acquires-nest-for-3-2-billion/>. Acesso em: 15/6/2014.

TSAI, C.-W.; LAI, C.-F.; CHIANG, M.-C.; YANG, L. T. Data Mining for Internet of Things: A Survey. **IEEE Communications Surveys Tutorials**, v. 16, n. 1, p. 77–97, 2014.

TUCIC, M.; PAVLOVIC, R.; PAPP, I.; SARIC, D. Networking layer for unifying distributed smart home entities. In: Telecommunications Forum Telfor, 22, 2014, Belgrade. **Anais ...** Belgrade: IEEE, 2014.

UDA, UPnP device architecture version 1.0, July 2006, document version 1.0.1. **UPnP Forum**, 2006. Disponível em <http://www.upnp.org/resources/documents/CleanUPnPDA101-20031202s.pdf> Acesso em: 15/6/2014.

UPNP FORUM SDKs**, UPnP Forum,** 2006. Disponível em: <http://upnp.org/sdcps-and-certification/resources/sdks/>. Acesso em: 15/6/2014.

UPNP FORUM, S. **Standardized DCPs & Certification**. Disponível em <http://upnp.org/sdcps-and-certification/>. Acesso em: 25/8/2015.

UPNP FORUM, W. PAPER. Understanding Universal Plug and Play white paper. **UPnP Forum,** 2006. Disponível em <www.upnp.org/download/UPNP_understandingUPNP.doc>. Acesso em: 22/12/2014.

UPNP FORUM. UPnP Forum Invites Organizations to Use UPnP+ Certification. **Marketwire**, 2014. Disponível em: <http://www.marketwired.com/press-release/upnp-forum-invites-organizations-to-use-upnp-certification-1969142.htm>. Acesso em: 22/12/2014.

VALTCHEV, D.; FRANKOV, I. Service gateway architecture for a smart home. **IEEE Communications Magazine**, v. 40, n. 4, p. 126–132, 2002.

VARCHOLA, M.; DRUTAROVSKÝ, M. ZigBee based home automation wireless sensor network. **Acta Electrotechnica et Informatica**, v. 7, p. 4, 2007.

VAXEVANAKIS, K.; ZAHARIADIS, T.; VOGIATZIS, N. A Review on Wireless Home Network Technologies. **SIGMOBILE Mobile Computer Communication Review**, v. 7, n. 2, p. 59–68, 2003.

WACKS, K. Home systems standards: achievements and challenges. **IEEE Communications Magazine**, v. 40, n. 4, p. 152–159, 2002.

WANG, H.; GE, Z.; LI, T. E2-DLNA: An Improved Energy-Efficiency Mechanism for DLNA. **Journal of Networks**, v. 9, n. 8, p. 2072–2078, 2014.

WANG, S.; XU, Z.; CAO, J.; ZHANG, J. A middleware for web service-enabled integration and interoperation of intelligent building systems. **Automation in Construction**, v. 16, n. 1, p. 112–121, 2007.

WEISER, M. Some computer science issues in ubiquitous computing. **Communications of ACM**, v. 36, n. 7, p. 75–84, 1993.

WEISER, M. The computer for the 21st century. **SIGMOBILE Mobile Computer Communication Review**, v. 3, n. 3, p. 3–11, 1999.

WILSON, C.; HARGREAVES, T.; HAUXWELL-BALDWIN, R. Smart Homes and Their Users: A Systematic Analysis and Key Challenges. **Personal Ubiquitous Computing**, v. 19, n. 2, p. 463–476, 2015.

WOOD, G.; NEWBOROUGH, M. Energy-use information transfer for intelligent homes: Enabling energy conservation with central and local displays. **Energy and Buildings**, v. 39, n. 4, p. 495–503, 2007.

XU, D.; THOMAS, L.; KENT, M.; MOUELHI, T.; LE TRAON, Y. A model-based approach to automated testing of access control policies. In: Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, 17, 2012, New York. **Anais ...** SACMAT New York, NY, USA: ACM, 2012.

YAN, X.; BO, W. A Security Extension to LonWorks/LonTalk Protocol. **International Journal of Digital Content Technology & its Applications**, v. 7, n. 6, p. 790, 2013.

YIM, H.-J.; LEE, K.-C. A DLNA-Based Efficient Mechanism for Interconnecting Mobile Devices in Ubiquitous Environments. In: Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications, 16, 2012, Washington. **Proceedings ...** Washington, DC, USA: IEEE Computer Society, 2012.

ZHANG, Y.; YU, R.; XIE, S.; et al. Home M2M Networks: Architectures, Standards, and QoS Improvement. **IEEE Communications Magazine**, v. 49, n. 4, p. 44–52, 2011.

ZHAO, X.; WANG, L.; GU, D. UPnP Extensions for Public IPv4 Sharing in IPv6 Environment. In: 2010 Sixth International Conference on Networking and Services (ICNS), 6, 2010, Cancun, Mexico. **Anais ...** Cancun: IEEE, 2010.

ZIGBEE ALLIANCE. *ZigBee Technology*. **ZIGBEE**, 2014. Disponível em: <www.zigbee.org\/About\/AboutTechnology\/ZigBeeTechnology.aspx>. Acesso em: 25 out. 2014.

APÊNDICE A

Código na classe Main do Sistema de Iluminação.

```
void *MicroStackChain;
void *UPnPmicroStack;
HANDLE ILib_IPAddressMonitorTerminator;
HANDLE ILib_IPAddressMonitorThread;
DWORD ILib_MonitorSocketReserved;
WSAOVERLAPPED ILib_MonitorSocketStateObject;
SOCKET ILib_MonitorSocket;
/* Variables */
int numOfDevicesSwitchPower=5;
int numOfDevicesDimming=5;
int numOfDevicesSwitchPowerDimming=5;
int UPnPSwitchPowerValueSwitch[5] = {0,0,0,0,0};
int UPnPSwitchPowerValueDimming[5] = {0,0,0,0,0};
int UPnPSwitchPowerValueSwitchDimming[5] = {0,0,0,0,0};
unsigned char UPnPDimmingLoadLevelTargetSwitch[5] = {0,0,0,0,0};
unsigned char UPnPDimmingLoadLevelTargetDimming[5] = {0,0,0,0,0};
unsigned char UPnPDimmingLoadLevelTargetSwitchDimming[5] = {0,0,0,0,0};
unsigned char UPnPDimmingLoadLevelTargetMinLevel=0;
FILE *fp;
time_t current_time;
char* c_time_string;
int i;

//Initialize
void UPnPDimming_GetLoadLevelStatus(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPDimming_GetLoadLevelStatus(%u): %i \r\n",
Device,UPnPSwitchPowerValueDimming[Device]);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimming_GetLoadLevelStatus(%u): %i \r\n",
Device,UPnPSwitchPowerValueDimming[Device]);
		fclose(fp);
	}
	UPnPResponse_Dimming_GetLoadLevelStatus(upnptoken,UPnPSwitchPowerValueDimming[Device]);
}
void UPnPDimming_GetLoadLevelTarget(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPDimming_GetLoadLevelTarget(%u): %i \r\n",
Device,UPnPSwitchPowerValueDimming[Device]);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimming_GetLoadLevelTarget(%u): %i \r\n",
Device,UPnPSwitchPowerValueDimming[Device]);
		fclose(fp);
	}
	UPnPResponse_Dimming_GetLoadLevelTarget(upnptoken,UPnPSwitchPowerValueDimming[Device]);
}

void UPnPDimming_GetMinLevel(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPDimming_GetMinLevel(%u): %i \r\n", Device, UPnPDimmingLoadLevelTargetMinLevel);
```

```
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimming_GetMinLevel(%u): %i \r\n", Device,
UPnPDimmingLoadLevelTargetMinLevel);
		fclose(fp);
	}
	UPnPResponse_Dimming_GetMinLevel(upnptoken,UPnPDimmingLoadLevelTargetMinLevel);
}

void UPnPDimming_SetLoadLevelTarget(UPnPSessionToken upnptoken,unsigned char NewLoadLevelTarget,unsigned
char Device){
	printf("Invoke: UPnPDimming_SetLoadLevelTarget(%u,%u);\r\n", NewLoadLevelTarget, Device);
	UPnPSwitchPowerValueDimming[Device]=NewLoadLevelTarget;
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimming_SetLoadLevelTarget(%u,%u);\r\n", NewLoadLevelTarget, Device);
		fclose(fp);
	}
	UPnPResponse_Dimming_SetLoadLevelTarget(upnptoken);
}

void UPnPDimming2_GetLoadLevelStatus(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPDimming2_GetLoadLevelStatus(%u): %u\r\n", Device,
UPnPSwitchPowerValueSwitchDimming[Device]);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimming2_GetLoadLevelStatus(%u): %u\r\n", Device,
UPnPSwitchPowerValueSwitchDimming[Device]);
		fclose(fp);
	}
	UPnPResponse_Dimming2_GetLoadLevelStatus(upnptoken,UPnPSwitchPowerValueSwitchDimming[Device]);
}

void UPnPDimming2_GetLoadLevelTarget(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPDimming2_GetLoadLevelTarget(%u): %u \r\n", Device,
UPnPSwitchPowerValueSwitchDimming[Device]);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimming2_GetLoadLevelTarget(%u): %u \r\n", Device,
UPnPSwitchPowerValueSwitchDimming[Device]);
		fclose(fp);
	}
	UPnPResponse_Dimming2_GetLoadLevelTarget(upnptoken,UPnPSwitchPowerValueSwitchDimming[Device]);
}

void UPnPDimming2_GetMinLevel(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPDimming2_GetMinLevel(%u): %u \r\n", Device, UPnPDimmingLoadLevelTargetMinLevel);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimming2_GetMinLevel(%u): %u \r\n", Device,
UPnPDimmingLoadLevelTargetMinLevel);
		fclose(fp);
	}
	UPnPResponse_Dimming2_GetMinLevel(upnptoken,UPnPDimmingLoadLevelTargetMinLevel);
}
```

```
void UPnPDimming2_SetLoadLevelTarget(UPnPSessionToken upnptoken,unsigned char NewLoadLevelTarget,unsigned
char Device){
	printf("Invoke: UPnPDimming2_SetLoadLevelTarget(%u,%u);\r\n", NewLoadLevelTarget, Device);
	UPnPSwitchPowerValueSwitchDimming[Device]=NewLoadLevelTarget;
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimming2_SetLoadLevelTarget(%u,%u);\r\n", NewLoadLevelTarget, Device);
		fclose(fp);
	}
	UPnPResponse_Dimming2_SetLoadLevelTarget(upnptoken);
}

void UPnPSwitchPower2_GetStatus(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPSwitchPower2_GetStatus(%u): %i \r\n", Device,
UPnPSwitchPowerValueSwitchDimming[Device]);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPSwitchPower2_GetStatus(%u): %i \r\n", Device,
UPnPSwitchPowerValueSwitchDimming[Device]);
		fclose(fp);
	}
	UPnPResponse_SwitchPower2_GetStatus(upnptoken,UPnPSwitchPowerValueSwitchDimming[Device]);
}

void UPnPSwitchPower2_GetTarget(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPSwitchPower2_GetTarget(%u): %i \r\n", Device,
UPnPSwitchPowerValueSwitchDimming[Device]);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPSwitchPower2_GetTarget(%u): %i \r\n", Device,
UPnPSwitchPowerValueSwitchDimming[Device]);
		fclose(fp);
	}
	UPnPResponse_SwitchPower2_GetTarget(upnptoken,UPnPSwitchPowerValueSwitchDimming[Device]);
}

void UPnPSwitchPower2_SetTarget(UPnPSessionToken upnptoken,int newTargetValue,unsigned char Device){
	printf("Invoke: UPnPSwitchPower2_SetTarget(%d,%u);\r\n", newTargetValue, Device);
	UPnPSwitchPowerValueSwitchDimming[Device]=newTargetValue;
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPSwitchPower2_SetTarget(%d,%u);\r\n", newTargetValue, Device);
		fclose(fp);
	}
	UPnPResponse_SwitchPower2_SetTarget(upnptoken);
}

void UPnPSwitchPower_GetStatus(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPSwitchPower_GetStatus(%u): %i \r\n", Device, UPnPSwitchPowerValueSwitch[Device]);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPSwitchPower_GetStatus(%u): %i \r\n", Device,
UPnPSwitchPowerValueSwitch[Device]);
		fclose(fp);
	}
	UPnPResponse_SwitchPower_GetStatus(upnptoken,UPnPSwitchPowerValueSwitch[Device]);
```

```
}

void UPnPSwitchPower_GetTarget(UPnPSessionToken upnptoken,unsigned char Device){
	printf("Invoke: UPnPSwitchPower_GetTarget(%u): %i \r\n", Device, UPnPSwitchPowerValueSwitch[Device]);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPSwitchPower_GetTarget(%u): %i \r\n", Device,
UPnPSwitchPowerValueSwitch[Device]);
		fclose(fp);
	}
	UPnPResponse_SwitchPower_GetTarget(upnptoken,UPnPSwitchPowerValueSwitch[Device]);
}

void UPnPSwitchPower_SetTarget(UPnPSessionToken upnptoken,int newTargetValue,unsigned char Device){
	printf("Invoke: UPnPSwitchPower_SetTarget(%d,%u);\r\n", newTargetValue, Device);
	UPnPSwitchPowerValueSwitch[Device]=newTargetValue;
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPSwitchPower_SetTarget(%d,%u);\r\n", newTargetValue, Device);
		fclose(fp);
	}
	UPnPResponse_SwitchPower_SetTarget(upnptoken);
}
// Group functions
void UPnPGetDevicesList_GetDeviceList(UPnPSessionToken upnptoken){
	printf("Invoke: UPnPGetDevicesList_GetDeviceList();\r\n");
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPGetDevicesList_GetDeviceList();\r\n");
		fclose(fp);
	}
	UPnPResponse_GetDevicesList_GetDeviceList(upnptoken,"Sample String");
}

void UPnPDimmingAll_GetLoadLevelStatus(UPnPSessionToken upnptoken){
	printf("Invoke: UPnPDimmingAll_GetLoadLevelStatus();\r\n");
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimmingAll_GetLoadLevelStatus();\r\n");
		fclose(fp);
	}
	//Loop all devices
	for(i=0; i<numOfDevicesDimming; i++){
		UPnPDimming_GetLoadLevelStatus(upnptoken,i);
	}
	for(i=0; i<numOfDevicesSwitchPowerDimming; i++){
		UPnPDimming2_GetLoadLevelStatus(upnptoken,i);
	}
	UPnPResponse_DimmingAll_GetLoadLevelStatus(upnptoken,250);
}

void UPnPDimmingAll_GetLoadLevelTarget(UPnPSessionToken upnptoken){
	printf("Invoke: UPnPDimmingAll_GetLoadLevelTarget();\r\n");
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimmingAll_GetLoadLevelTarget();\r\n");
```

```
		fclose(fp);
	}
	//Loop all devices
	for(i=0; i<numOfDevicesDimming; i++){
		UPnPDimming_GetLoadLevelTarget(upnptoken,i);
	}
	for(i=0; i<numOfDevicesSwitchPowerDimming; i++){
		UPnPDimming2_GetLoadLevelTarget(upnptoken,i);
	}
	UPnPResponse_DimmingAll_GetLoadLevelTarget(upnptoken,250);
}

void UPnPDimmingAll_GetMinLevel(UPnPSessionToken upnptoken){
	printf("Invoke: UPnPDimmingAll_GetMinLevel();\r\n");
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimmingAll_GetMinLevel();\r\n");
		fclose(fp);
	}
	UPnPResponse_DimmingAll_GetMinLevel(upnptoken,UPnPDimmingLoadLevelTargetMinLevel);
}

void UPnPDimmingAll_SetLoadLevelTarget(UPnPSessionToken upnptoken,unsigned char NewLoadLevelTarget){
	printf("Invoke: UPnPDimmingAll_SetLoadLevelTarget(%u);\r\n", NewLoadLevelTarget);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPDimmingAll_SetLoadLevelTarget(%u);\r\n", NewLoadLevelTarget);
		fclose(fp);
	}
	//Loop all devices
	for(i=0; i<numOfDevicesDimming; i++){
		UPnPDimming_SetLoadLevelTarget(upnptoken,NewLoadLevelTarget,i);
	}
	for(i=0; i<numOfDevicesSwitchPowerDimming; i++){
		UPnPDimming2_SetLoadLevelTarget(upnptoken,NewLoadLevelTarget,i);
	}
	UPnPResponse_DimmingAll_SetLoadLevelTarget(upnptoken);
}

void UPnPSwitchPowerAll_GetStatus(UPnPSessionToken upnptoken){
	printf("Invoke: UPnPSwitchPowerAll_GetStatus();\r\n");
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPSwitchPowerAll_GetStatus();\r\n");
		fclose(fp);
	}
	//Loop all devices
	for(i=0; i<numOfDevicesSwitchPower; i++){
		UPnPSwitchPower_GetStatus(upnptoken, i);
	}
	for(i=0; i<numOfDevicesSwitchPowerDimming; i++){
		UPnPSwitchPower2_GetStatus(upnptoken, i);
	}
	UPnPResponse_SwitchPowerAll_GetStatus(upnptoken,1);
}

void UPnPSwitchPowerAll_GetTarget(UPnPSessionToken upnptoken){
	printf("Invoke: UPnPSwitchPowerAll_GetTarget();\r\n");
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
```

```
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPSwitchPowerAll_GetTarget();\r\n");
		fclose(fp);
	}
	//Loop all devices
	for(i=0; i<numOfDevicesSwitchPower; i++){
		UPnPSwitchPower_GetTarget(upnptoken, i);
	}
	for(i=0; i<numOfDevicesSwitchPowerDimming; i++){
		UPnPSwitchPower2_GetTarget(upnptoken, i);
	}
	UPnPResponse_SwitchPowerAll_GetTarget(upnptoken,1);
}

void UPnPSwitchPowerAll_SetTarget(UPnPSessionToken upnptoken,int newTargetValue){
	printf("Invoke: UPnPSwitchPowerAll_SetTarget(%d);\r\n", newTargetValue);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Invoke: UPnPSwitchPowerAll_SetTarget(%d);\r\n", newTargetValue);
		fclose(fp);
	}
	//Loop all devices
	for(i=0; i<numOfDevicesSwitchPower; i++){
		UPnPSwitchPower_SetTarget(upnptoken,newTargetValue,i);
	}
	for(i=0; i<numOfDevicesSwitchPowerDimming; i++){
		UPnPSwitchPower2_SetTarget(upnptoken,newTargetValue,i);
	}
	UPnPResponse_SwitchPowerAll_SetTarget(upnptoken);
}

void CALLBACK ILib_IPAddressMonitor(IN DWORD dwError, IN DWORD cbTransferred, IN LPWSAOVERLAPPED
lpOverlapped,
IN DWORD dwFlags){
	UNREFERENCED_PARAMETER( dwError );
	UNREFERENCED_PARAMETER( cbTransferred );
	UNREFERENCED_PARAMETER( lpOverlapped );
	UNREFERENCED_PARAMETER( dwFlags );

UPnPIPAddressListChanged(UPnPmicroStack);
	WSAIoctl(ILib_MonitorSocket, SIO_ADDRESS_LIST_CHANGE, NULL, 0, NULL, 0,
&ILib_MonitorSocketReserved, &ILib_MonitorSocketStateObject, &ILib_IPAddressMonitor);
}

DWORD WINAPI ILib_IPAddressMonitorLoop(LPVOID args){
	UNREFERENCED_PARAMETER( args );
	ILib_MonitorSocket = socket(AF_INET,SOCK_DGRAM,0);
	WSAIoctl(ILib_MonitorSocket, SIO_ADDRESS_LIST_CHANGE, NULL, 0, NULL, 0,
&ILib_MonitorSocketReserved, &ILib_MonitorSocketStateObject, &ILib_IPAddressMonitor);
	while(WaitForSingleObjectEx(ILib_IPAddressMonitorTerminator, INFINITE, TRUE) != WAIT_OBJECT_0);
	return 0;
}

DWORD WINAPI Run(LPVOID args){
	UNREFERENCED_PARAMETER( args );
	getchar();
	closesocket(ILib_MonitorSocket);
	SetEvent(ILib_IPAddressMonitorTerminator);
	WaitForSingleObject(ILib_IPAddressMonitorThread, INFINITE);
	CloseHandle(ILib_IPAddressMonitorTerminator);
	ILibStopChain(MicroStackChain);
	return 0;
}
```

```
int _tmain(int argc, _TCHAR* argv[]){
	DWORD ptid = 0;
	DWORD ptid2 = 0;
	UNREFERENCED_PARAMETER( argc );
	UNREFERENCED_PARAMETER( argv );
	_CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);
	MicroStackChain = ILibCreateChain();
	UPnPmicroStack = UPnPCreateMicroStack(MicroStackChain, "Network Light System (2CRH0Y1)", "Dimmable
Light", "Light Bulb", "Simple Ligh", "28ea94b1-44bc-408a-8980-cdc6fa2f40df", "0000001", 1800, 0);
	UPnPFP_Dimming_GetLoadLevelStatus =
(UPnP_ActionHandler_Dimming_GetLoadLevelStatus)&UPnPDimming_GetLoadLevelStatus;
	UPnPFP_Dimming_GetLoadLevelTarget =
(UPnP_ActionHandler_Dimming_GetLoadLevelTarget)&UPnPDimming_GetLoadLevelTarget;
	UPnPFP_Dimming_GetMinLevel =
(UPnP_ActionHandler_Dimming_GetMinLevel)&UPnPDimming_GetMinLevel;
	UPnPFP_Dimming_SetLoadLevelTarget =
(UPnP_ActionHandler_Dimming_SetLoadLevelTarget)&UPnPDimming_SetLoadLevelTarget;
	UPnPFP_Dimming2_GetLoadLevelStatus =
(UPnP_ActionHandler_Dimming2_GetLoadLevelStatus)&UPnPDimming2_GetLoadLevelStatus;
	UPnPFP_Dimming2_GetLoadLevelTarget =
(UPnP_ActionHandler_Dimming2_GetLoadLevelTarget)&UPnPDimming2_GetLoadLevelTarget;
	UPnPFP_Dimming2_GetMinLevel =
(UPnP_ActionHandler_Dimming2_GetMinLevel)&UPnPDimming2_GetMinLevel;
	UPnPFP_Dimming2_SetLoadLevelTarget =
(UPnP_ActionHandler_Dimming2_SetLoadLevelTarget)&UPnPDimming2_SetLoadLevelTarget;
	UPnPFP_DimmingAll_GetLoadLevelStatus =
(UPnP_ActionHandler_DimmingAll_GetLoadLevelStatus)&UPnPDimmingAll_GetLoadLevelStatus;
	UPnPFP_DimmingAll_GetLoadLevelTarget =
(UPnP_ActionHandler_DimmingAll_GetLoadLevelTarget)&UPnPDimmingAll_GetLoadLevelTarget;
	UPnPFP_DimmingAll_GetMinLevel =
(UPnP_ActionHandler_DimmingAll_GetMinLevel)&UPnPDimmingAll_GetMinLevel;
	UPnPFP_DimmingAll_SetLoadLevelTarget =
(UPnP_ActionHandler_DimmingAll_SetLoadLevelTarget)&UPnPDimmingAll_SetLoadLevelTarget;
	UPnPFP_GetDevicesList_GetDeviceList =
(UPnP_ActionHandler_GetDevicesList_GetDeviceList)&UPnPGetDevicesList_GetDeviceList;
	UPnPFP_SwitchPower_GetStatus =
(UPnP_ActionHandler_SwitchPower_GetStatus)&UPnPSwitchPower_GetStatus;
	UPnPFP_SwitchPower_GetTarget =
(UPnP_ActionHandler_SwitchPower_GetTarget)&UPnPSwitchPower_GetTarget;
	UPnPFP_SwitchPower_SetTarget =
(UPnP_ActionHandler_SwitchPower_SetTarget)&UPnPSwitchPower_SetTarget;
	UPnPFP_SwitchPower2_GetStatus =
(UPnP_ActionHandler_SwitchPower2_GetStatus)&UPnPSwitchPower2_GetStatus;
	UPnPFP_SwitchPower2_GetTarget =
(UPnP_ActionHandler_SwitchPower2_GetTarget)&UPnPSwitchPower2_GetTarget;
	UPnPFP_SwitchPower2_SetTarget =
(UPnP_ActionHandler_SwitchPower2_SetTarget)&UPnPSwitchPower2_SetTarget;
	UPnPFP_SwitchPowerAll_GetStatus =
(UPnP_ActionHandler_SwitchPowerAll_GetStatus)&UPnPSwitchPowerAll_GetStatus;
	UPnPFP_SwitchPowerAll_GetTarget =
(UPnP_ActionHandler_SwitchPowerAll_GetTarget)&UPnPSwitchPowerAll_GetTarget;
	UPnPFP_SwitchPowerAll_SetTarget =
(UPnP_ActionHandler_SwitchPowerAll_SetTarget)&UPnPSwitchPowerAll_SetTarget;

// All evented state variables MUST be initialized before UPnPStart is called.
UPnPSetState_GetDevicesList_DeviceList(UPnPmicroStack, "Sample String");
UPnPSetState_DimmingAll_LoadLevelStatus(UPnPmicroStack, 250);
UPnPSetState_SwitchPowerAll_Status(UPnPmicroStack, 1);
// All evented state variables MUST be initialized before UPnPStart is called.
UPnPSetState_Dimming_LoadLevelStatus(UPnPmicroStack, 250);
// All evented state variables MUST be initialized before UPnPStart is called.
UPnPSetState_Dimming2_LoadLevelStatus(UPnPmicroStack, 250);
UPnPSetState_SwitchPower2_Status(UPnPmicroStack, 1);
// All evented state variables MUST be initialized before UPnPStart is called.
UPnPSetState_SwitchPower_Status(UPnPmicroStack, 1);
```

```
	printf("MicroStack 1.0 - Network Light System (2CRH0Y1) \r\n\r\n");
	/* Obtain current time as seconds elapsed since the Epoch. */
	current_time = time(NULL);
	if (current_time == ((time_t)-1)){
		printf("Failure to compute the current time.");
		return EXIT_FAILURE;
	}
	/* Convert to local time format. */
	c_time_string = ctime(&current_time);
	if (c_time_string == NULL){
		printf("Failure to convert the current time.");
		return EXIT_FAILURE;
	}
	/* Print to stdout. */
	printf("Current time is %s", c_time_string);
	fp=fopen("LOG.txt", "a");
	if (fp == NULL) {
		printf("Can't open output file!\n");
	}else{
		fprintf(fp, "Start Simulation from %s \n", c_time_string);
		fclose(fp);
	}

	CreateThread(NULL, 0, &Run, NULL, 0, &ptid);

	ILib_IPAddressMonitorTerminator = CreateEvent(NULL, TRUE, FALSE, NULL);
	ILib_IPAddressMonitorThread = CreateThread(NULL, 0, &ILib_IPAddressMonitorLoop, NULL, 0, &ptid2);
	ILibStartChain(MicroStackChain);
	return 0;
}
```