

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ALYSTON CLAYDSON VERÍSSIMO DO NASCIMENTO

Estudo de Caso: Arquitetura UPnP

São Luís
2012

ALYSTON CLAYDSON VERÍSSIMO DO NASCIMENTO

Estudo de Caso: Arquitetura UPnP

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Carlos de Salles Soares Neto.

São Luís
2012

Nascimento, Alyston.

Estudo de Caso: Arquitetura UPnP / Alyston Nascimento- 2012

88.p.

Impresso por Computador (Fotocópia).

Orientador: Prof. Dr. Carlos de Salles Soares Neto.

Monografia (Graduação)- Universidade Federal do Maranhão, Curso de Ciência da Computação, 2012.

1.Redes de acesso local 2.Padrão UPnP. 3.DLNA. I.Título.

CDU 004.725.5

ALYSTON CLAYDSON VERÍSSIMO DO NASCIMENTO

Estudo de Caso: Arquitetura UPnP

Monografia apresentada ao Curso de
Ciência da Computação da Univer-
sidade Federal do Maranhão, como
parte dos requisitos necessários para
a obtenção do grau de Bacharel em
Ciência da Computação.

Aprovado em 09/07/2012

BANCA EXAMINADORA

Carlos de Salles Soares Neto

Prof. Carlos de Salles Soares Neto (Orientador)

Doutor em Informática
Universidade Federal do Maranhão

Maria Auxiliadora Freire

Prof^a Maria Auxiliadora Freire

Mestre em Ciência de Engenharia
Universidade Federal do Maranhão

Portela

Prof. Carlos Eduardo Portela Serra de Castro

Mestre em Informática
Universidade Federal do Maranhão

Aos meus pais e irmã.

As minhas lindas e maravilhosas filhas.

Aos amigos, pelo apoio e companheirismo.

RESUMO

O termo UPnP é a evolução do termo PnP que significa "Ligar e Usar", uma tecnologia para conexão dinâmica de dispositivos em uma rede para compartilhar os recursos. Universal Plug and Play é um conjunto de protocolos de rede de computadores, idealizado inicialmente pela Microsoft e passado para o Fórum UPnP. Os objetivos da arquitetura UPnP são para conexão direta e simplificada, sem configuração, de equipamentos em pequenas redes domésticas ou comerciais. A tecnologia tem o nome de "Universal Ligar e Usar" por se tratar de ligar o equipamento em qualquer meio físico de rede, sem qualquer tipo de configuração. A arquitetura UPnP conecta os equipamentos de forma automática, dessa forma um ambiente inteligente seria um lugar o qual vários serviços e sensores estariam presentes e disponível ao usuário. Este trabalho visa fazer uma pesquisa sobre a arquitetura UPnP, como funciona o processo de conexão dos dispositivos de maneira automática, uma pesquisa sobre as atuais aplicações e as possíveis aplicações para o uso desta tecnologia.

Palavras-chaves: Redes de acesso local. Padrão UPnP. DLNA. Protocolos de comunicação em redes.

ABSTRACT

The term UPnP is the evolution of the term PnP it means "Plug and Play", a technology for dynamic connection of devices on a network to share resources. Universal Plug and Play is a set of computer network protocols, initially conceived by Microsoft and passed to the UPnP Forum. The goals of UPnP architecture are for direct connection and simplified, no configuration of equipment in small business or home networks. The technology is called "Universal Plug and Play" because it is turning on the machine in any medium physical network without any configuration. The UPnP architecture connects the equipment automatically, so an intelligent environment would be a place where the various services and sensors would be present and available to the user. This work aims to do a search on the UPnP architecture, how the process of connecting devices in an automatic way, a survey of current applications and potential applications for this technology.

Keywords: Local area networks. UPnP standard. DLNA. Network communication protocols.

AGRADECIMENTOS

Primeiramente a Deus, fonte de vida. A todos os meus parentes, pelo encorajamento e apoio.

Ao professor Carlos Salles pela orientação, amizade e principalmente, pela paciência, apoio incondicional, sem a qual este trabalho não se realizaria.

Aos meus colegas de turma pelo apoio, encorajamento e companheirismo.

Aos professores do Departamento de Informática pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o meu enriquecimento pessoal e profissional.

“As pessoas que vencem neste mundo são as que procuram as circunstâncias de que precisam e, quando não as encontram, as criam.”

Bernard Shaw - Filósofo

Lista de Siglas:

AL - Alternate Local

API - Application Programming Interface

ARP - Address Resolution Protocol

AV - Áudio e Vídeo

DCP - Device Control Protocol

DHCP - Dynamic Host Configuration Protocol

DLNA - Digital Living Network Alliance

DMC - Digital Media Controller DMP - Digital Media Player

DMR - Digital Media Renderer

DMS - Digital Media Server

DNS - Domain Name System

DTCP-IP - Digital Transmission Content Protection for Internet Protocol

DTD - Document Type Definition

FDDI - Fiber Distributed Data Interface

FNC - Federal Networking Council

FTP - File Transfer Protocol

GENA - General Event Notification Architecture

GNU - General Public License

HTTP - HyperText Transfer Protocol

HTTPMU - HyperText Transfer Protocol Multicast

HTTTPU - HyperText Transfer Protocol Unicast

IAB - Internet Advisory Board

IANA - Internet Assigned Numbers Authority

ICMP - Internet Control Message Protocol

IEEE - Institute of Electrical and Electronics Engineers

IETF - Internet Engineering Task Force

IGMP - Internet Group Management

IMAP - Internet Message Access Protocol

InterNIC - Internet Network Information Center

IP - Internet Protocol

IrDA - Infrared Data Association

IRTF - Internet Research Task Force

ISA - Industry Standard Architecture

ISDN - Integrated Services Digital Network

ITU - International Telecommunication Union

LAN - Local Area Network

MIME - Multipurpose Internet Mail Extensions

NAT - Network Address Translation

NFS - Network File System

OSI - Open Systems Interconnection

PCI - Peripheral Component Interconnect

PnP - Plug and Play

POP3 - Post Office Protocol 3

PPP - Point-to-Point Protocol

RARP - Reverse Address Resolution Protocol

RF - Rádio Frequência

RFC - Request for Comments

RPC - Remote Procedure Calls

SCSI - Small Computer System Interface

SDK - Software Development Kit

SGML - Standard Generalized Markup Language

SLIP - Serial Line Internet Protocol

SMTP - Simple Mail Transfer Protocol

SNMP - Simple Network Management Protocol

SOAP - Simple Object Access Protocol

SSDP - Simple Service Discovery Protocol

TCP - Transmission Control Protocol

TFTP - Trivial File Transfer Protocol

TLI - Transport Layer Interface

UDP - User Datagram Protocol

UFMA - Universidade Federal do Maranhão

UIC - UPnP Implementer's Corporation

UPnP - Universal Plug and Play

URI - Uniform Resource Identifiers

URL - Uniform Resource Locator

URN - Uniform Resource Name

UUID - Universal Unique Identifier

W3C - World Wide Web Consortium

WAN - Wide Area Network

XML - eXtensible Markup Language

Lista de Figuras

2.1	Camadas do TCP/IP (COMER,1995)	21
2.2	Pilha de Protocolos nas Camadas TCP/IP (COMER,1995)	22
3.1	Dependências entre as Fases do UPnP (JERONIMO e WEAST,2003).	45
3.2	Pilha Protocolos Ponto a Ponto (JERONIMO e WEAST,2003)	46
3.3	Pilha Protocolos Multicast (JERONIMO e WEAST,2003)	47
4.1	Algoritmo de Endereçamento dos Dispositivos UPnP (JERONIMO e WEAST,2003).	53
4.2	Exemplo do inicio um document XML, para um dispositivo	55
4.3	Pilha de Protocolos GENA	59
5.1	Fases desenvolvimento usando Intel SDK para UPnP (INTEL,2002)	64
5.2	SDK Service Author aba State Variable	65
5.3	SDK Service Author aba Action	66
5.4	SDK Service Author SwitchPower.xml	66
5.5	SDK Device Builder LampadaIP	67
5.6	SDK Device Builder Adicionar Serviço	67
5.7	SDK Device Builder Generate Stack	68
5.8	SDK Device Spy Teste da LampadaIP	69
5.9	SDK Device Spy Teste Serviço da LampadaIP	69
5.10	Arquitetura mínima para UPnP A/V (JERONIMO e WEAST,2003)	70
5.11	Diagrama de Sequencia Mensagens UPNP A/V (JERONIMO e WEAST,2003)	71
5.12	Página de Apresentação do MediaTomb	73
5.13	Pilha de Protocolos DLNA (SONY,2012)	75

5.14 Certificação DLNA (SONY,2012)	76
--	----

Sumário

Lista de Figuras	9
1 INTRODUÇÃO	15
1.1 Motivação	15
1.1.1 Por que estudar UPnP?	15
1.1.2 Alguns cenários de uso da Arquitetura UPnP.	16
1.2 Objetivos	17
1.2.1 Gerais	18
1.2.2 Específicos	18
1.3 Estrutura do trabalho	18
2 Fundamentação Teórica	19
2.1 Uniform Resource Identifiers (URI)	19
2.1.1 Uniform Resource Locators (URL)	19
2.1.2 Uniform Resource Names (URN)	20
2.1.3 Camada de Rede	20
2.1.4 Camada de Inter-Rede	21
2.1.5 Camada de Transporte	22
2.1.6 Camada de Aplicação	23
2.2 IP Multicast	23
2.3 HyperText Transfer Protocol (HTTP)	24
2.3.1 Conexões	25
2.3.2 Métodos	25
2.3.3 Cabeçalhos de Mensagens	26

2.4	Protocolos: HTTPU e HTTPMU	26
2.4.1	O cabeçalho de solicitação MX	27
2.4.2	Cabeçalho Geral Parâmetro "S" (Sequencer)	28
2.4.3	Cabeçalho Geral Parâmetro "AL" (Alternate Local)	28
2.4.4	O Parâmetro "*" (Request URI)	29
2.5	Extensible Markup Language (XML)	29
2.5.1	Documentos XML	31
2.5.2	A sintaxe de XML	31
3	Arquitetura UPnP	36
3.1	Introdução	36
3.1.1	Serviços suportado pela arquitetura UPnP	38
3.2	Fórum UPnP	40
3.2.1	Os comitês do Fórum UPnP	41
3.3	Componentes de uma rede UPnP	43
3.3.1	Dispositivos UPnP	43
3.3.2	Serviços UPnP	44
3.3.3	Ponto de Controle	44
3.4	As Fases UPnP	44
3.4.1	Dependências entre as Fases do UPnP	45
3.5	A Pilha de Protocolos UPnP	46
3.5.1	Comunicação Ponto a Ponto	46
3.5.2	Comunicação Multicast	46
4	Pilha de Protocolos UPnP	48
4.1	Endereçamento dos Dispositivos UPnP	48
4.1.1	DHCP	49
4.1.2	AutoIP	50

4.1.3	Algoritmo de endereçamento dos dispositivos UPnP	52
4.2	Descoberta dos Dispositivos UPnP	52
4.2.1	Protocolo SSPD	53
4.3	Descrição dos Dispositivos UPnP	55
4.4	Controle dos Dispositivos UPnP	56
4.4.1	Remote Procedure Calls (RPC)	56
4.4.2	SOAP	57
4.5	Eventos em Dispositivos UPnP	58
4.5.1	Eventos em um sistema Distribuído	58
4.5.2	General Event Notification Architecture (GENA)	58
4.6	Apresentação	61
5	Cenários de Uso	63
5.1	Desenvolvimento de Dispositivos UPnP	63
5.2	Exemplo Simples de Criação de um Dispositivo UPnP: LâmpadaIP	65
5.2.1	Primeiro passo: Gerar o arquivo de Especificação do Dispositivo	65
5.2.2	Segundo passo: Gerar e Compilar o código em C#	66
5.2.3	Terceiro passo: Testar o dispositivo recém criado.	68
5.3	Arquitetura UPnP em Áudio/Vídeo	69
5.3.1	Cenário de Uso: Reprodução de Áudio	70
5.4	Aplicação em Automação Residencial	71
5.5	Servidor de Mídia UPnP - Media Tomb	72
5.5.1	Recursos Suportados pelo Servidor de Mídia	73
5.5.2	Teste da Página de Apresentação Media Tomb	74
5.6	DLNA - Digital Living Network Alliance	74
5.6.1	Característica DLNA	75
5.6.2	Dispositivos Envolvidos	75

5.6.3	Certificação de Dispositivos	76
6	CONCLUSÃO	77
6.1	Avaliação do Trabalho	78
6.2	Trabalhos Futuros	79
	REFERÊNCIAS BIBLIOGRÁFICAS	80
	ANEXOS	82

1 INTRODUÇÃO

A arquitetura UPnP, surgiu há mais de dez anos, com a evolução do conceito de "ligar e usar". Em outros termos é possível ligar o equipamento, sem nenhuma necessidade de configuração por parte do usuário. Hoje comercialmente não está mais favorável aos novos usuários comprar equipamentos e ter que configurar para poder usar, os consumidores querem facilidade. A arquitetura UPnP vem estender o conceito de ligar e usar, antes dentro do computador para fora dele, ou seja, nas redes de computadores e dispositivos, principalmente residenciais, não só de computadores, mas também ter dispositivos domésticos com tecnologia embarcada, chamados de equipamentos inteligentes, compartilhando recursos e facilitando o uso dos dispositivos.

1.1 Motivação

A principal motivação para a elaboração deste trabalho é o uso crescente da tecnologia aplicada à arquitetura UPnP e nas aplicações em diversas áreas, principalmente em sistemas residenciais, entender como é a aplicação do padrão UPnP em projetos, como ele simplifica a produção de aplicações em redes locais, para diferentes contextos, tais como aplicações em redes locais em geral e redes de uso doméstico.

1.1.1 Por que estudar UPnP?

A principal preocupação da arquitetura UPnP é poder entender e desenvolver novas ferramentas para tornar o uso compartilhado dos dispositivos residenciais, tornar isso rápido e voltado à aplicação, sem a necessidade de estar resolvendo problemas de conectividade e sim voltado para a solução do problema no nível de aplicação. A arquitetura usa tecnologia de protocolos amplamente aplicada e aprovada, e, o que é mais importante, a arquitetura.

1.1.2 Alguns cenários de uso da Arquitetura UPnP.

Essa seção apresenta alguns cenários possíveis com o uso da arquitetura UPnP aplicada. Com a criação de dispositivos compatíveis facilitada, integração simples, os equipamentos compatíveis possuem descoberta automática dos equipamentos e o controle, isso facilita a integração, podendo melhorar de diversas maneiras, como compartilhar os recursos para tornar a vidas dos usuários melhor.

1.1.2.1 Home Network

Observando o uso da arquitetura UPnP nas residências, percebe-se a necessidade de acesso a dados e serviços distribuídos e a necessidade de autoconfiguração, pois é desejável que o usuário final interaja o mínimo possível na configuração de equipamentos. A criação de novos equipamentos com essas características faz surgir um novo tipo de equipamento, podemos ver algumas aplicações desse novo tipo de equipamento.

Uma pessoa chegou em sua casa depois de um dia difícil no trabalho e entrou pela porta da frente. Ele aciona um interruptor na parede que, para a maioria de nós, apenas acende somente um ponto de luz. Com o UPnP, esse interruptor faz muito mais. Ele é apenas um serviço UPnP cujo estado é definido por uma variável chamada "position". Quando é acionada a variável do interruptor, muda para "on" e o ponto de controle previamente recebe a notificação do evento, inicializa um programa de configuração e aciona outro dispositivo que executa os seguintes comandos.

1. O ponto de luz da sala acende.
2. O ar-condicionado é ligado na temperatura pré-definida.
3. A secretária eletrônica começa a reproduzir as novas mensagens.
4. A televisão é ligada e sintonizada no canal de notícias, com o volume do som no nível agradável.
5. A persiana da janela motorizada sobe.

Da mesma forma, quando você desativa o interruptor, colocando-o na posição "off", o ponto de controle recebe o evento da variável modificado para "off", é executado pelo ponto de controle os seguintes comandos:

1. O ar-condicionado é desligado
2. A Televisão é desligada.
3. A persiana da janela desce, para oferecer privacidade.
4. Todas as luzes da casa são apagadas.

Esse é somente um exemplo, totalmente possível, usando dispositivos inteligentes e equipamentos compatíveis com o padrão UPnP.

1.1.2.2 Configuração Automática dos Roteadores Domésticos

O procedimento de abrir portas em roteadores para possibilitar o uso de programas de comunicação e troca de arquivos é difícil e demora, para a maioria dos usuários. Os usuários avançados já têm certa facilidade em configurar. É necessário saber que portas os programas usa e também tem que configurar o recurso de NAT (Network Address Translation) dos roteadores . Essas tarefas exigem um nível médio de conhecimento de redes, internet e informática, um pouco além do que os usuários iniciantes estão acostumados. Os modems/routers compatíveis com UPnP podem fazer a configuração do NAT Transverse automaticamente. Basta habilitar a opção nos roteadores, os programas aplicativos, como o Skype ¹ ou o eMule² solicitam ao roteador abrir as portas necessárias automaticamente.

O programa Skype faz isso sem nenhuma configuração, mas o eMule precisa ser configurado manualmente para fazer a solicitação ao roteador.

1.2 Objetivos

Este trabalho tem como objetivo pesquisar sobre tecnologia envolvida na arquitetura UPnP, no intuito de facilitar o compartilhamento de recursos em uma rede de computadores.

¹Skype programa muito usado para comunicação entre pessoas usando VoIP (Voice over Internet Protocol)

²eMule programa usado para compartilhamento de arquivos através do uso cliente/servidor

1.2.1 Gerais

Estudar o padrão UPnP, as fases especificadas pelo padrão UPnP para descoberta e disponibilização de serviços, descrição de dispositivo e disponibilização e acesso a serviços e eventos vinculados a um determinado dispositivo.

1.2.2 Específicos

Pesquisar as ferramentas de desenvolvimento disponíveis, verificar a complexidade de implementação para o Padrão UPnP, pesquisar sobre os protocolos padrões usados, tais como TCP, UDP e HTTP, como eles resolvem cada problema de conectividade dentro da arquitetura UPnP.

1.3 Estrutura do trabalho

O restante do trabalho esta estruturado da seguinte forma:

- No **Capítulo 2** contém os conceitos básicos usados na arquitetura UPnP, incluindo como URI, TCP, IP, IP Multicast, HTTP, HTTPU, HTTPMU, XML; protocolos básicos e conceitos usados para o estudo da solução empregada na Arquitetura UPnP.
- No **Capítulo 3** contém a descrição da Arquitetura UPnP componentes, conceitos, histórico, comitê regulatório e trata das fases conceituais da Arquitetura UPnP.
- No **Capítulo 4** contém o detalhamento da implementação usada para a arquitetura UPnP, todos os protocolos usados e detalhes da solução em cada fase conceitual a pilha de protocolos UPnP.
- No **Capítulo 5** contém os cenários de uso, exemplo de dispositivos simples da tecnologia, quais aplicações utiliza e alguns exemplos conceituais do uso da Tecnologia UPnP, por exemplo o padrão DLNA.
- no **Capítulo 6** contém a conclusão do trabalho apresentando opinião, a avaliação do trabalho, as dificuldades encontradas na arquitetura UPnP e trabalhos futuros. Ao final temos as referências bibliográfica usada em todo o trabalho.

2 Fundamentação Teórica

A arquitetura UPnP usa vários protocolos e conceitos, para tornar mais fácil o compartilhamento de recursos dentro de uma pequena rede existe vários conceitos aplicados nessa arquitetura, protocolos, recursos, padrões, dentre outros. Nesta seção vamos revisar algum dos assuntos teóricos muito importantes para formar um fundamento teórico básico para o entendimento dos capítulos posteriores, principalmente a partir do capítulo três arquitetura UPnP.

2.1 Uniform Resource Identifiers (URI)

Um Uniform Resource Identifiers (URI) prove um simples e extensível meio para identificar um recurso é usado para classificar um localização, um nome, ou ambos (RFC3986,2005). A internet possui vários recursos disponíveis, mas para podemos acessá-los temos que diferenciar estes recursos, isso se chama de Uniform Resource Identifiers (URI), Identificador Uniforme de Recursos. URI é uma cadeia de caracteres compacta usada para identificar ou denominar um recurso na internet. O principal propósito nesta identificação e representar os recursos de uma forma padrão, usando os protocolos específicos usados na internet. URI são identificadas em grupos definidos em uma sintaxe específica e protocolos associados, essa sintaxe é definida pela RFC3986 de 2005. A URI é composta de Uniform Resource Name (URN) e Uniform Resource Locator URL, (URN) é usado para identificar um recurso através de um nome único, já URL identifica um recurso através de sua localização, ou seja, através do endereço. Podemos fazer uma analogia a uma pessoa, um indivíduo possui seu nome isso é o seu URN, o endereço da sua casa é o seu URL, ambos identificam esse indivíduo de forma única no mundo.

2.1.1 Uniform Resource Locators (URL)

A Uniform Resource Locator (URL) faz parte da URI que identifica um recurso, especificando a sua localização. Muitos esquemas de URL são nomeados após protocolos. Por exemplo, URLs usando o esquema HTTP começam com `http://` e especificar

para utilizar o protocolo HTTP para recuperar o recurso. URL's que começam com `ftp://` especificar para utilizar o protocolo FTP para recuperar o recurso. (RFC3986,2005). A URL mostra como chegar a esse recurso em específico dentro da internet, ou seja, mostra o endereço do recurso.

2.1.2 Uniform Resource Names (URN)

Um URN é um rótulo usando para dar nome a um recurso, que é obrigatório para permanecer único no domínio global e persistente, ou seja, não muda nunca, mesmo quando o recurso deixar de existir ou se tornar indisponível o URN não poderá ser usado, ou seja, nunca poderá ser reutilizado por outro recurso. (RFC3986,2005). A URN descreve de maneira única o nome de um recurso específico dentro da internet.

TCP/IP (Transmission Control Protocol/Internet Protocol) é um conjunto de protocolos, onde dois dos mais importantes. O protocolo IP, base da estrutura de comunicação da Internet é um protocolo baseado no paradigma de chaveamento de pacotes.

Os protocolos TCP/IP podem ser utilizados sobre qualquer estrutura de rede, seja ela simples como uma ligação ponto-a-ponto ou uma rede de pacotes complexa. Como exemplo, pode-se empregar estruturas de rede como Ethernet, Token-Ring, FDDI, PPP, X.25 ¹, Frame-Relay, barramentos SCSI, enlaces de satélite, ligações telefônicas discadas e várias outras como meio de comunicação do protocolo TCP/IP (COMER,1995).

A arquitetura TCP/IP, assim como OSI realiza a divisão de funções do sistema de comunicação em estruturas de camadas. Em TCP/IP as camadas são mostradas na figura 2.1:

2.1.3 Camada de Rede

A camada de rede é responsável pelo envio de datagramas construídos pela camada Inter-Rede. Essa camada realiza também o mapeamento entre um endereço de identificação de nível Inter-rede para um endereço físico ou lógico do nível de Rede. A

¹X.25 é um conjunto de protocolos padronizado pela União Internacional de Telecomunicações (ITU) para redes de longa distância e que usam o sistema telefônico ou Integrated Service Digital Network (ISDN) como meio de transmissão.

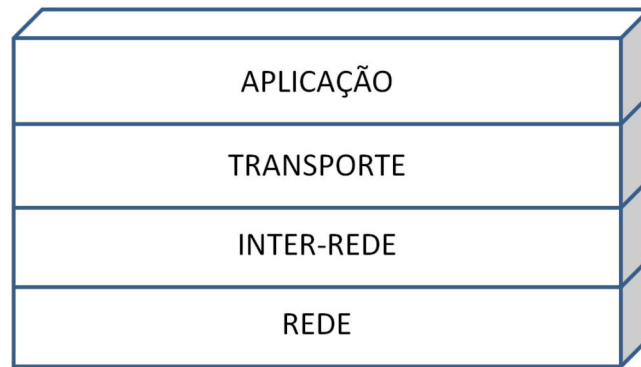


Figura 2.1: Camadas do TCP/IP (COMER,1995)

camada Inter-Rede é independente do nível de Rede. (COMER,1995).

Em resumo essa camada envia e recebe os dados para o meio físico, ou seja, e a camada que transmite as informações de dispositivo para o outro.

Alguns protocolos existentes nesta camada são:

- Protocolos com estrutura de rede própria (X.25, Frame-Relay)
- Protocolos de Enlace OSI (Ethernet, Token-Ring, SLIP)
- Protocolos de Nível Físico (V.24, X.21)
- Protocolos de barramento de alta-velocidade (SCSI)

2.1.4 Camada de Inter-Rede

Essa camada realiza a comunicação entre máquinas vizinhas através do protocolo IP. Para identificar cada máquina e a própria rede onde essas estão situadas, é definido um identificador, chamado endereço IP, que é independente de outras formas de endereçamento que possam existir nos níveis inferiores. No caso de existir endereçamento nos níveis inferiores é realizado um mapeamento para possibilitar a conversão de um endereço IP em um endereço deste nível (COMER,1995).

Os protocolos existentes nesta camada são:

- Protocolo de transporte de dados: IP - Internet Protocol
- Protocolo de controle e erro: ICMP - Internet Control Message Protocol

- Protocolo de controle de grupo de endereços: IGMP - Internet Group Management Protocol
- Protocolos de controle de informações de roteamento

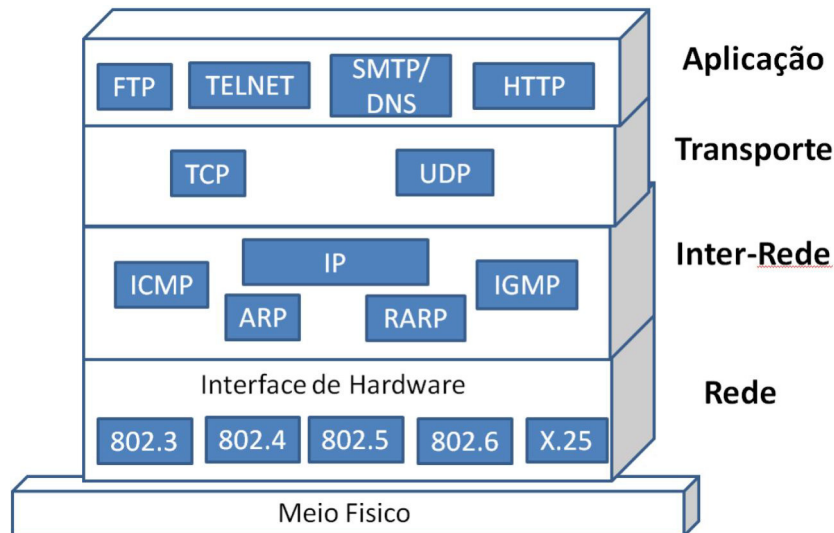


Figura 2.2: Pilha de Protocolos nas Camadas TCP/IP (COMER,1995)

O Nível Inter-Rede compreende principalmente os protocolos IP e ICMP e IGMP (Internet Group Management Protocol). Os protocolos ARP e RARP são pertencentes na verdade aos dois níveis, Inter-rede e Rede, pois realizam funções com informações de ambos.

2.1.5 Camada de Transporte

Essa camada reúne os protocolos que realizam as funções de transporte de dados fim-a-fim, ou seja, considerando apenas a origem e o destino da comunicação, sem se preocupar com os elementos intermediários. A camada de transporte possui dois protocolos que são o UDP (User Datagram Protocol) e TCP (Transmission Control Protocol). O protocolo UDP realiza apenas a multiplexação para que várias aplicações possam acessar o sistema de comunicação de forma coerente. O protocolo TCP realiza, além da multiplexação, uma série de funções para tornar a comunicação entre origem e destino mais confiável. São responsabilidades do protocolo TCP: o controle de fluxo, o controle de erro, a seqüência e a multiplexação de mensagens. A camada de transporte oferece para o nível de aplicação um conjunto de funções e procedimentos para acesso ao sistema de

comunicação de modo a permitir a criação e a utilização de aplicações de forma independente da implementação. Desta forma, as interfaces socket ou TLI (ambiente Unix) e Winsock (ambiente Windows) fornecem um conjunto de funções-padrão para permitir que as aplicações possam ser desenvolvidas independentemente do sistema operacional no qual rodarão.

2.1.6 Camada de Aplicação

A camada de aplicação reúne os protocolos que fornecem serviços de comunicação ao sistema ou ao usuário. Pode-se separar os protocolos de aplicação em protocolos de serviços básicos ou protocolos de serviços para o usuário: Protocolos de serviços básicos, que fornecem serviços para atender as próprias necessidades do sistema de comunicação TCP/IP: DNS, BOOTP, DHCP Protocolos de serviços para o usuário: FTP, HTTP, Telnet, SMTP, POP3, IMAP, TFTP, NFS, RealAudio, Gopher, Archie, Finger, SNMP e outros.

2.2 IP Multicast

IP Multicast é a transmissão de um datagrama IP para um "grupo de hosts", representado por um conjunto de zero ou mais hosts identificados por um único endereço IP de destino. Um datagrama deste tipo é entregue a todos os membros pertencentes a este grupo com a mesma confiabilidade "best-efforts" existente em pacotes IP unicast, ou seja, não há garantia que o pacote atinja o seu destino intacto a todos os membros do grupo destino ou que cheguem à mesma ordem em que foram enviados. (RFC1112, 1989).

IP Multicast é uma solução antiga para o problema de enviar uma única mensagem para vários remetentes, hoje existe outras implementações, as mais usadas são Internet Relay Chat e PSYC ². A conferência por Multicast IP foi apresentada inicialmente em maior escala quando foi usada para transmitir várias sessões do 23º IETF em Março de 1992 para pesquisadores e observadores ao redor do mundo. Desde então, seções especiais da IETF tem sido transmitidas por Multicast via MBONE ³ e redes Multicast

²PSYC é o protocolo, um dos mais rápidos, ainda extensível baseado em texto, fornecendo uma infraestrutura de troca de mensagens para a conversação humana e de também para troca de uso social de arquivos binários. <http://about.psync.eu/>

³MBONE ou "Multicast Backbone" é uma rede virtual dentro da camada física da Internet, e teve

privadas.

Características relativas à associação de hosts a grupos

- A associação a um grupo é dinâmica; hosts podem participar ou abandoná-los a qualquer momento.
- Não há restrição quanto ao posicionamento geográfico ou número de membros em um grupo de hosts.
- Um host pode ser membro de um ou mais grupos ao mesmo tempo.
- Um host não precisa ser membro de um grupo para enviar datagramas a ele. Grupos
- Um grupo de hosts pode ser permanente ou transiente. Um grupo permanente tem um IP bem conhecido. É o endereço e não a associação que é permanente; a qualquer momento um grupo permanente pode ter um número qualquer de membros, até mesmo zero.
- Os endereços multicast que não são reservados para grupos permanentes estão disponíveis para atribuições dinâmicas para grupos transientes que existem somente enquanto houver hosts membros.

2.3 HyperText Transfer Protocol (HTTP)

O protocolo de transferência utilizado em toda a Word Wide Web é o HTTP. Esse protocolo especifica as mensagens que os clientes podem enviar aos servidores e que respostas eles receberão. Cada interação consiste em uma solicitação ASCII, seguida por uma resposta RFC822 semelhante ao MIME(Multipurpose Internet Mail Extensions). Todos os clientes e todos os servidores devem obedecer a esse protocolo. Ele é definido na RFC2616. (TANENBAUM,2003).

início em Março de 1992 durante uma reunião da IETF na qual foi transmitido o evento ao vivo para 30 pessoas que participaram remotamente. O Mbone foi chamado assim por Steve Casner pesquisador da University of Southern California Information Sciences Institute.

Método	Descrição
GET	Solicita a leitura de uma página
HEAD	Solicita a leitura de um cabeçalho de página Web
PUT	Solicita o armazenamento de uma página Web
POST	Acrescenta a um recurso
DELETE	Remove a página Web
TRACE	Ecoa a solicitação recebida
CONNECT	reservado para uso futuro
OPTIONS	Consulta certas opções

Tabela 2.1: Alguns Métodos HTTP

2.3.1 Conexões

O cliente normalmente estabelece uma conexão TCP com o servidor pela porta 80, essa é a porta padrão. O uso do TCP tem varias vantagens, não é preciso se preocupar com as mensagens perdidas, duplicadas longas, ou ter que se preocupar com as confirmações, todos esses problemas são tratados pela implementação do TCP. (TANENBAUM,2003).

2.3.2 Métodos

O protocolo HTTP tem seu grande uso na Web, mas ele foi criado de um modo mais geral que o uso na web, visando principalmente a aplicações orientadas a objetos, logo são aceitas operações chamadas por métodos, diferente de uma simples solicitação de página na Web. (TANENBAUM,2003).

O método GET solicita que o servidor envie a página ou objeto. GET nomearq HTTP/1.1 Onde nomearq identifica o arquivo a ser buscado e 1.1 é a versão do protocolo que está sendo usado. Listado na tabela 2.1 alguns métodos do Protocolo HTTP.

Para toda a solicitação obtém uma resposta que consiste em uma linha de status, esses códigos estão listados na tabela 2.2. Os códigos 1xx são raramente usados. Os códigos 2xx significa que a solicitação foi tratada com sucesso. Os códigos 3xx informam que o cliente que ele deve procurar em outro servidor, redirecionamento. Os códigos 4xx significa que a solicitação falhou devido a um erro do cliente. Os códigos

Código	Significado	Exemplos
1xx	Informação	100 = o servidor concorda em atender a solicitação
2xx	Sucesso	200 = requisição bem sucedida; 204 = sem conteúdo
3xx	Redirecionamento	301 = a página foi movida
4xx	Erro do Cliente	403 = página proibida; 404 = página não encontrada
5xx	Erro Servidor	500 = erro interno no servidor; 503 = tente de novo depois

Tabela 2.2: Status Resposta HTTP

5xx significa que o servidor tem um problema seja por erro interno ou uma sobrecarga. (TANENBAUM,2003).

2.3.3 Cabeçalhos de Mensagens

Na linha de solicitação pode ser seguida de linha adicionais, essas linhas são chamadas de cabeçalhos de solicitação, essas linhas podem ser comparadas com parâmetros de uma chamada de procedimento, já as respostas também podem ter esses cabeçalhos, chamados de cabeçalhos de resposta. (TANENBAUM,2003).

2.4 Protocolos: HTTPU e HTTPMU

HTTP é um protocolo de solicitação/resposta genérica que é feita através de TCP, ou seja, a comunicação é feita por conexão. Existem muitos outros tipos de comunicação que não podem ser eficientemente modeladas utilizando TCP, tais como o envio de uma única mensagem para um grupo de receptores, o host teria que enviar a mesma mensagem a todos os destinatários individualmente. O que é necessário é uma maneira de manter os benefícios de HTTP, sobre UDP unicast ou multicast na camada de transporte.(JERONIMO e WEAST, 2003).

HTTP sobre UDP unicast, HTTPU, proporciona os benefícios de HTTP, juntamente com a simplicidade do UDP. Com HTTPU, um host pode enviar uma mensagem HTTP formatado para outro host sem o custo de configurar uma conexão TCP. HTTP sobre multicast UDP, HTTPMU, permite o envio de mensagens HTTP, mas para vários destinatários simultaneamente. HTTPMU permite um modelo de comunicação em grupo

utilizando HTTP-estilo request /mensagens de resposta.(JERONIMO e WEAST, 2003).

Embora existam benefícios claros para ser capaz de enviar HTTP sobre UDP, isso introduz alguns problemas que precisam ser resolvidos:

- Se um host envia mensagem para muitos receptores, todos eles podem responder ao mesmo tempo, talvez possa sobrecarregar a capacidade do host em processar as respostas.
- Se um host envia vários pedidos para vários destinatários, deve coincidir com as respostas, recebidos de forma assíncrona, com o pedido original.
- Cabeçalho HTTP do local é usado para identificar uma URI em uma mensagem de resposta, como quando redirecionando o cliente para um novo URI. HTTP suporta apenas um cabeçalho único local em mensagens de resposta. Novos modelos de comunicação ativado por HTTP sobre UDP pode exigir vários locais a serem especificados em uma resposta.

HTTPU e HTTPMU introduzir três novos cabeçalhos HTTP para abordar estas questões: o cabeçalho do pedido MX, o parâmetro S do cabeçalho geral, e o parâmetro AL do cabeçalho geral. (JERONIMO e WEAST, 2003).

2.4.1 O cabeçalho de solicitação MX

O cabeçalho de solicitação MX é usado para ajudar a dividir, as respostas ao longo do tempo dos pedidos HTTPMU para que eles não retornem a uma taxa maior do que o solicitante pode processar. O cabeçalho MX especifica o número máximo de segundos que um multicast UDP HTTP o recurso tem que esperar antes de enviar uma resposta iniciado por um pedido de multicast anterior.(JERONIMO e WEAST, 2003).

Se um recurso tem uma única resposta a um pedido, que gera um número aleatório entre 0 e MX que representa o número de segundos que o recurso espera antes de enviar a sua resposta.

Se um recurso tem várias respostas para o pedido, ele deve enviar esses recursos distribuídos ao longo do intervalo [0 .. MX].(JERONIMO e WEAST, 2003)

2.4.2 Cabeçalho Geral Parâmetro "S" (Sequencer)

Como o UDP é sem conexão, não há nenhuma maneira para um cliente HTTPU / HTTPMU para associar uma resposta de um servidor com um pedido que o cliente tenha enviado anteriormente. O cabeçalho S fornece essa capacidade. O cabeçalho S contém um URI absoluto que é único em todo o namespace URI. Quando um cabeçalho S é enviado por um cliente logo após um pedido HTTPU / HTTPMU, o servidor simplesmente retorna o mesmo valor na resposta.(JERONIMO e WEAST, 2003).

Se um cliente recebe várias respostas com o cabeçalho mesmo S, então o cliente pode assumir que todas as respostas referem-se à mesma solicitação. No entanto, porque as respostas são transportados por UDP, não há garantias de entrega dos pacotes para as respostas. Múltiplas respostas com o cabeçalho com o mesmo S podem aparecer em qualquer ordem. Os clientes não são obrigados a enviar o cabeçalho S, é puramente para a conveniência do cliente de acordo com as respostas aos pedidos.(JERONIMO e WEAST, 2003).

2.4.3 Cabeçalho Geral Parâmetro "AL" (Alternate Local)

Cabeçalho HTTP de resposta de localização permite um único URI a ser devolvido em uma resposta. Por exemplo, para HTTP 201 (Created) respostas, o cabeçalho Location especifica o novo recurso que foi criado com o pedido. O valor de cabeçalho local consiste de uma única URL absoluta, como se segue:

Localização: `http://www.intoast.com/index.htm`

Usando HTTPMU e HTTPU, existem muitos casos em que um recurso precisa fornecer informações de localização usando múltiplas URI's. O cabeçalho de Localização padrão não é suficiente neste caso. O cabeçalho da AL é um novo cabeçalho para HTTPMU / HTTPU que tem a mesma semântica como o cabeçalho local, mas pode conter múltiplos URI's. O conteúdo de um cabeçalho de AL está ordenado. Se um cabeçalho de localização e um cabeçalho AL estão incluídos na mesma mensagem, então o URI no cabeçalho local poderá ser tratada como se fosse à primeira entrada no cabeçalho AL.(JERONIMO e WEAST, 2003).

2.4.4 O Parâmetro "*" (Request URI)

Além de especificar novos cabeçalhos, outra modificação é necessária para HTTP padrão para enviar o HTTP sobre multicast IP. No protocolo HTTP, o cliente envia uma solicitação para o servidor que consiste de um método de solicitação, um campo URI de solicitação, e a versão do protocolo HTTP. Por exemplo,

```
GET HTTP/1.1 http://www.intoast.com/index.htm
```

O campo URI de solicitação de uma solicitação HTTP especifica o recurso a ser recuperado. No exemplo anterior, o cliente faz pedido para uma página web na www.intoast.com para o host. No entanto, quando usado com uma solicitação HTTP de difusão seletiva, o URI de solicitação tem um significado ligeiramente diferente. Em vez de especificar um recurso a ser recuperado, o Request-URI especifica para quem o pedido é destinado. HTTPU e HTTPMU usar "*", que significa "para todos aqueles que estão me ouvindo esse endereço IP e a porta." (JERONIMO e WEAST, 2003).

Por exemplo,

```
M-SEARCH "*" HTTP/1.1
```

2.5 Extensible Markup Language (XML)

Em meados de 1996, um grupo de trabalho do Word Wide web Consortium (W3C), organização encarregada em desenvolver e manter a maior parte dos padrões da Web deu início ao desenvolvimento de uma linguagem de marcação para solucionar as limitações da HTML e que tivesse o poder e a generalidade da SGML⁴ (Standart Generalized Markup Language) e que, ao mesmo tempo, fosse fácil de ser implementada na Web. (LIGHT,1999).

Esse trabalho deu origem a XML (Extensible Markup Language), uma linguagem de marcação extensível que possuía os seguintes objetivos de projeto (BRAY, 2008).

XML foi desenvolvida principalmente para transportar e armazenar dados, independente da plataforma, fazendo uma comparação HTML tem a finalidade de mostrar

⁴SGML - Linguagem Padrão de Marcação Generalizada que fornece um esquema de marcação simples, independente de plataforma e extremamente flexível.

os dados. XML é uma complementação para o HTML, enquanto HTML mostra os dados, XML, armazena e transfere os dados, pode-se fazer páginas dinâmicas, juntando as duas ferramentas. Em um documento de formato XML os dados são descritos em texto. Sabemos bem que um formato de texto não é a solução mais econômica para armazenar dados, mas muitas vezes é a mais eficaz. Melhor, porque é mais fácil trabalhar com um texto, do que trabalhar com os formatos binários típicos de muitas bases de dados. (HEITLINGER, 2001). A XML surgiu para solucionar alguns problemas encontrados na HTML. Essa é uma aplicação da SGML que possui um conjunto de tags⁵ pré-definidas. Essas tags servem basicamente para apresentar o conteúdo de documentos em páginas Web e não dão nenhum sentido a eles. XML, entretanto, possibilita: (FURGERI, 2001).

- A criação de tags conforme as necessidades do usuário, possibilitando o surgimento de linguagens baseadas em XML.
- A visualização do mesmo documento de diferentes formas, por meio da utilização de folhas de estilos, ordenando e filtrando as informações, segundo certos critérios;
- A estrutura criada pelo documento XML permite que ferramentas baseadas em bancos de dados possam consultar e processar seu conteúdo;
- Ferramentas desenvolvidas para a manipulação de XML facilitam a criação de documentos, permitindo que o desenvolvedor se dedique apenas ao conteúdo das informações, já que a estrutura é controlada pela ferramenta.
- A XML traz benefícios tanto para aquele que produz a informação como para aquele que a recebe. A maneira como a estrutura de um documento XML é formada possibilita que os processos sejam automatizados por meio de software que reconhecem as tags do documento XML, integrando o site da Internet com o software interno de uma empresa, reduzindo custos e tornando os processos mais eficientes, tanto para o produtor dos documentos quanto para quem vai ler seu conteúdo;
- A XML é um padrão permanente, já que o conteúdo de seu documento XML pode ser lido e atualizado pela ferramenta que estiver sendo utilizada;
- Os browsers, apoiados por linguagens de programação, podem fazer grande parte do processamento das informações; o conteúdo do documento pode ser manipulado

⁵Tag significa: Etiqueta, Rótulo; uma maneira de identificar alguma coisa.

e reorganizado; cálculos podem ser realizados para gerar novos conteúdos instantaneamente, o que proporciona a geração de novos documentos;

- Os recursos fornecidos pela XML podem ser usados para criar uma rede de conhecimento, interligando documentos com informações complementares, mesmo que eles estejam em lugares diferentes na Web, ou ainda, que pertençam a diferentes empresas.

2.5.1 Documentos XML

XML é uma linguagem usada para descrever e manipular documentos estruturados. Todo documento escrito em XML deve ser criado de forma concisa e clara (FURGERI, 2001). Um documento XML é um arquivo texto que contém elementos, atributos, dados de caractere, comentários, entidades, instruções de processamento, entre outros. Um documento para ser considerado um documento XML deve ser bem formado. Para isto ele deve atender as seguintes regras definidas pelo W3C (BRAY, 2008)

- Ter a declaração XML na primeira linha do documento,
- Incluir um ou mais elementos;
- Ter exatamente um elemento raiz que deve conter todos os outros elementos;
- O nome da marca de fim (`</CLIENTE>`) deve coincidir com a marca de início (`<CLIENTE>`);
- As tags devem ser aninhadas adequadamente, isto é, não pode haver sobreposição;
- Cada atributo de um elemento deve ter um nome exclusivo;
- Cada uma das entidades analisadas, referidas dentro do documento, deve ser bem formada.

2.5.2 A sintaxe de XML

Um documento XML possui blocos de construção, os blocos de construção podem ter: elementos, atributos, cadeias de caracteres, instruções de processamento, comentários, seções CDATA, referências, entre outros. Nas seções a seguir é explanado de forma resumida cada uma delas.

2.5.2.1 Elementos

Um elemento XML é a unidade mais básica do documento. Ele é composto de uma tag de abertura, o conteúdo e uma tag de fechamento (HEITLINGER, 2001). A tag de abertura é formada pelo nome da tag escrito entre os sinais de menor que (<) e maior que (>). Um elemento geralmente é terminado com uma tag de fechamento, formada pelo mesmo nome precedido por uma barra inclinada, e delimitado pelos sinais de menor que e maior que (< /nome>). (CASTRO, 2001). O conteúdo de um elemento pode ser outros elementos ou texto e até ambos. Exemplo: <nome>José Silva</nome>

O nome da tag é criado pelo próprio desenvolvedor e deve descrever a finalidade do elemento e seu conteúdo em particular. O nome da tag deve começar com uma letra ou um sublinhado, seguido por letras, dígitos, sublinhados, hífens ou pontos, mas não pode haver espaços em branco no nome.

2.5.2.2 Elemento Raiz (root)

O primeiro elemento de um documento XML é chamado de elemento raiz (root). Ele engloba todos os outros elementos do documento e sua utilização é obrigatória. Esse elemento pode ser comparado à raiz de um disco rígido utilizado nos computadores, a partir do qual todas as informações são armazenadas em pastas (CASTRO, 2001).

2.5.2.3 Elemento vazio

Os elementos que não possuem conteúdo são conhecidos como elementos vazios. Normalmente, eles são incluídos no documento pelo valor de seus atributos. Um elemento vazio é formado pelo nome do elemento seguido de caracter ”/” entre os sinais de menor que (<) e maior que (>).

2.5.2.4 Atributos

Os atributos podem ser utilizados para fornecer propriedades especiais para as tags presentes no documento XML. Eles podem ser utilizados de diversas maneiras e são muito úteis para os documentos XML, pois podem anexar informações aos elementos.

Cada atributo possui um nome e um valor. Os nomes seguem as mesmas regras

dos nomes de elementos. O nome é separado do valor pelo caracter de igualdade (=). O valor do atributo deve aparecer entre aspas ou apóstrofos.

Os elementos podem ter zero ou mais atributos na tag de início ou na tag de elemento vazio.

Mas o mesmo atributo não pode aparecer mais de uma vez no mesmo elemento.

2.5.2.5 Instruções de Processamento

São elementos especiais adicionados ao documento XML para informar certos detalhes às ferramentas que realizarão sua interpretação. A ideia é permitir que o documento XML transmitisse alguma informação para o software que irá interpretar seu conteúdo. Uma instrução de processamento é delimitada pelos caracteres `<? e ?>`.

2.5.2.6 Declaração do Tipo do Documento (DTD)

Trata-se de uma instrução adicionada ao documento XML que deverá ser utilizado para verificar se todas as tags usadas estão seguindo certos padrões. Essa declaração do tipo do documento é opcional. Ela apontará para um documento usado para descrever quais tags podem aparecer no documento, o número e a sequência que podem aparecer, seus atributos e os valores que esses atributos podem possuir.

2.5.2.7 Declaração XML

A declaração XML, se existir, é a primeira linha do documento. Ela identifica o documento como um documento XML. A declaração também relaciona a versão da XML usada no documento, o padrão de codificação de caracteres e se o documento possui ou não entidades externas. Para isto ele define os atributos `version`, `standalone` e `encoding`, respectivamente. O código abaixo é um exemplo de uma declaração XML:

2.5.2.8 Comentários XML

Assim como nas linguagens de programação, XML também possui o recurso de comentário que facilita a compreensão dos itens presentes no documento. Sempre que necessário, os comentários podem ser utilizados para adicionar notas e observações no

documento e não serão levados em consideração pelo interpretador do documento XML. Para inserir comentários em um documento, delimite-os entre "`<!-- -->`":

2.5.2.9 Entidades

Uma entidade se refere a um caracter ou a um bloco de texto que será "importado" pelo documento XML toda vez que ele aparecer. Dessa forma, um interpretador do documento pode pesquisar a entidade referenciada e substituir seu conteúdo a partir do ponto em que foi encontrado. As entidades são inseridas no documento por meio de referências de entidade (o nome da entidade entre um caracter `&` e um sinal de ponto-e-vírgula). Para a aplicação, a referência de entidade é substituída pelo conteúdo dessa entidade. Se for considerado que foi definida uma entidade 'eua', que possui o valor 'Estados Unidos da América', ao ler a linha

```
<país>&br;< /país>
```

o processador XML a substituirá por

```
<país>Brasil< /país>
```

XML predefine entidades para os caracteres utilizados na marcação (sinais de menor e maior, aspas, entre outros). As entidades são usadas para substituir os caracteres no conteúdo do elemento ou do atributo. As entidades predefinidas são:

- `<`; para criar um sinal de menor que (`<`);
- `>`; para criar um sinal de maior que (`>`);
- `&`; para criar um caractere `&`;
- `&após;` para criar uma marcação de apóstrofo (`'`).
- `"`; para criar uma marcação de aspas (`"`).

2.5.2.10 Seção CDATA

As seções CDATA pode ocorrer com qualquer dados de caracteres, pois eles são usados para blocos de texto contendo caracteres que seriam interpretados como marcação. As seções CDATA começam com uma string "`< [CDATA [`" e terminam com uma string "`]] >`". (BRAY,2008).

Observações sobre seções CDATA: Uma seção CDATA não pode conter a seqüência (string) `]]>`, portanto, seções CDATA aninhadas não são permitidas.

Certifique-se de que não haja espaços ou quebras de linha dentro da string `]]>`.

3 Arquitetura UPnP

O UPnP é uma arquitetura de rede aberta e distribuída, usa protocolos padronizados altamente usados na internet, ele é independente de qualquer sistema operacional, linguagem de programação ou meio físico.

Nesta capítulo vai ser exposto, uma breve introdução, como e quando nasceu a arquitetura UPnP, os componentes de uma rede UPnP, quem regulamenta os padrões usados, os conceitos básicos de funcionamento usado, os protocolos usados como eles interagem e uma explicação básica do funcionamento de cada fase.

3.1 Introdução

O conceito de Plug and Play (PnP) surgiu para facilitar a conexão de dispositivos dentro de um computador pessoal. Antes os usuários tinham que instalar um programa específico fornecido pelo fabricante para configurar as placas que eram conectadas nos computadores através de barramentos, sem essa configuração a nova placa poderia entrar em conflito, usar a mesma faixa de endereçamento que outra placa já instalada, assim o computador não funcionava e outro programa fornecido pelo fabricante para o sistema operacional poder usar os recursos que a placa disponibilizava, ou seja, não era uma tarefa simples para pessoas sem um conhecimento básico de hardware de computadores e instalação e uso de software, essa tarefa se torna muito mais complicada para usuários comuns dos computadores pessoais.

A partir da necessidade de facilitar e padronizar a Intel criou o PnP (Plug and Play) e o incorporou ao barramento PCI, a Intel como principal fabricante de circuitos integrados de conexão entre dispositivos do computador pessoal utilizados nas placas principais dos computadores, deu o pontapé inicial para a padronização. Mas, somente muitos anos depois, a Microsoft com o sistema operacional Windows 95, trouxe suporte ao PnP, esta parte se faz muito importante para fechar o ciclo completo da instalação usando a tecnologia Plug and Play. O lançamento do PnP acelerou a demanda por computadores com Barramento PCI, rapidamente substituindo o barramento ISA como o mais usado nos

Data	Evento
Jan/99	Pardon UPnP Anunciando publicamente
Out/99	Fórum UPnP Formado Oficialmente
Jun/00	Arquitetura UPnP version 1 Finalizada
Jun/00	Microsoft Windows ME com Suporte a UPnP version 1
Jul/00	Intel lança open source UPnP SDK
Mai/01	UPnP version 1 toolkits Anunciado
Out/01	Microsoft Windows XP Com suporte UPnP version 1
Nov/01	Primeiro padrão para dispositivo UPnP publicado
Dez/01	Primeiro Dispositivo UPnP
Jan/02	Microsoft Windows CE com Suporte UPnP version 1

Tabela 3.1: Histórico UPnP

computadores pessoais, justamente pela facilidade embarcada no barramento PCI para época. O conceito "Plug na Play" foi criado a partir de 1985, mas sua popularidade se deu com o lançamento do Barramento PCI em 1990.

O exemplo do Plug an Play (PnP) de padronizar a interface das placas de periféricos com a placa principal foi a ideia base que se estendeu para fora do computador pessoal com o objetivo de conectar os próprios computadores sem a necessidade de nenhuma configuração adicional ou tipo de específico de mídia. Assim a Microsoft deu início a arquitetura UPnP (Universal Plug and Play).

O núcleo de arquitetura UPnP foi originalmente desenvolvido pela Microsoft foi passado para o Fórum UPnP na forma de especificação de Arquitetura UPnP. A especificação foi aprovada pelo Comitê Técnico do Fórum UPnP em 13 de junho de 2000. A Versão 1 da especificação enumera os principais protocolos UPnP e estabelece a base de trabalho que as comissões vão usar para desenvolver os seus dispositivos específicos. Na tabela 3.1 mostra a cronologia da arquitetura.

A arquitetura UPnP foi desenvolvida para conectar dispositivos de rede, como Computadores, equipamentos de entretenimento como televisão, estações de multimídia, vídeo games e outros tipos variados de aplicações inteligentes, que necessitam de compartilhamento de informações. Ele define um conjunto básico de padrões para todos os tipos dispositivos, convencionar e descrever os serviços que prestam. A arquitetura

UPnP utiliza padrões já existentes, como TCP/IP, HTTP, XML, em vez de modificar padrões já amplamente utilizados. A arquitetura é composta de um conjunto de protocolos padronizados onde cada dispositivo com UPnP habilitado pode fornecer métodos para a descoberta, controle e de transferência de dados entre dispositivos UPnP na mesma rede. A tecnologia UPnP pode ser suportada em qualquer sistema operacional ou plataforma de hardware e funciona com quase qualquer tipo de usuário, com qualquer tipo de mídia com ou sem fio, a escolha da plataforma e da mídia fica a cargo do desenvolvedor.

A arquitetura UPnP se baseia nas seguintes premissas (BOLZANI, 2004):

- **Padrões abertos:** baseado no protocolo IP e outros, sendo compatível com todos os equipamentos que utilizam este protocolo.
- **Escalabilidade:** permite reaproveitamento de equipamentos, sendo adquirido somente o necessário.
- **Plug and Play:** a ideia é apenas adquirir o equipamento e conecta-lo na rede e usar, sem configuração.
- **Leveza:** arquitetura leve, utilizada tanto em computadores pessoais como em dispositivos baseados em micro-controladores.
- **Integração:** fácil integração com dispositivos não IP, bem como de equipamentos de outros fabricantes.
- **Multiconfiguração:** pode operar em uma topologia ponto a ponto, sem a necessidade de um computador específico como gerenciador.

3.1.1 Serviços suportado pela arquitetura UPnP

Uma rede com dispositivos compatíveis com UPnP tem vários serviços e funcionalidades com o intuito de facilitar as interconexões entre os dispositivos, os quais vão prover serviços, como para os pontos de controle que vão usufruir dos serviços, abaixo algumas das funcionalidades e praticidades disponíveis em uma rede UPnP. (JERONIMO e WEAST, 2003).

3.1.1.1 Conectividade entre Dispositivos

A arquitetura UPnP define os protocolos para os dispositivos interagirem em uma rede, para compartilhar os recursos. A UPnP define como os dispositivos podem ser anunciados, quando é conectado ou desconectado da rede. Isso é feito de forma transparente, assim quando se publica os serviços, descobre-se outros dispositivos e serviços, envia-se eventos e controla-se dispositivos. (JERONIMO e WEAST, 2003).

3.1.1.2 Redes Ad-Hoc

As redes Ad-Hoc são formadas entre dispositivos sem a necessidade de equipamento de infraestrutura, ou seja, redes entre dispositivos sem roteadores, switches e hub.

Os dispositivos UPnP podem se juntar a uma rede de dispositivos dinamicamente, sem a necessidade de uma estrutura de serviços dedicada para a rede, ou seja, a rede não precisa de um servidor de DHCP para fornecer um endereço de IP para haver conectividade entre os dispositivos, na falta do Servidor de DHCP os próprios dispositivos compatíveis com UPnP automaticamente atribui seu próprio endereço de IP.

3.1.1.3 Redes Sem Configuração

Na arquitetura UPnP um usuário pode conectar um dispositivo compatível sem qualquer configuração adicional para poder usar os recursos disponíveis na rede. UPnP. (JERONIMO e WEAST, 2003).

Essa característica pode ser considerada muito importante, já que a facilidade em se conectar em uma rede sem se preocupar com configuração, torna acessível para a maioria dos usuários.

3.1.1.4 Arquiteturas Baseada em Padrões

A arquitetura UPnP é baseada em padrões abertos, incluindo as propostas existentes definidos pelo Internet Engineering Task Force (IETF) e World Wide Web Consortium (W3C) utilizando protocolos como o IP, TCP, UDP, HTTP, XML e SOAP. Aproveitando tecnologias existentes baseadas na Internet simplificando o projeto de dis-

positivos UPnP.

3.1.1.5 Independência de Plataforma

A Arquitetura UPnP é primeiramente um conjunto de protocolos e não é definido como Application Programming Interface API. O desenvolvimento de aplicações usando a arquitetura UPnP não necessita de uma plataforma específica, pode ser usado em qualquer plataforma independente do sistema operacional, linguagem ou hardware. UPnP. (JERONIMO e WEAST, 2003).

3.1.1.6 Independência na Mídia de Conexão

Tecnologia UPnP pode rodar em qualquer mídia, incluindo linha de telefones, linha de energia elétrica, Ethernet, RF e IEEE 1394.

3.1.1.7 Controle de dispositivos de forma manual ou automática

A arquitetura UPnP possibilita que os aplicativos rodando em um computador pessoal ou tablet ¹ controla de forma automática os dispositivos conectados em uma rede doméstica com UPnP habilitado, ou seja, pode-se criar programas para controlar os dispositivos sem a interferência de usuários ou o próprio usuário controlar usando uma interface administrativa baseada em HTML com qualquer aplicativo Browser.

3.2 Fórum UPnP

A arquitetura UPnP foi apresentado pela Microsoft Corporation no Consumer Electronics Show em Janeiro de 1999. A ideia foi originalmente apresentada pelas empresas Microsoft, Intel, Hewlett-Packard, Compaq, Dell e dentre outras, essa ideia foi considerada a próxima fase da Iniciativa Plug-and-Play introduzida pela Intel, Compaq, e Microsoft em 1992.

Tem como objetivo guiar a criação de padrões, o grupo de trabalho é formado por representantes da própria indústria. Hoje o Fórum UPnP é formado por represen-

¹É um dispositivo pessoal em formato de prancheta que pode ser usado para acesso a internet, organização pessoal, leitura de livros, jornais, etc.

tantes de mais de 550 empresas, incluindo os líderes de mercado no ramo de eletrônicos, computadores, automação residencial, segurança residencial, impressão, fotografia, rede de computadores e celulares (JERONIMO e WEAST, 2003).

As principais atividades do Fórum UPnP incluem:

- Definição de padrões de dispositivos baseados na arquitetura UPnP.
- Prover a certificação de dispositivos.
- Facilitar a promoção conjunta de membros UPnP.

As descrições dos dispositivos são documentos XML, baseados em um esquema de documentação que fornece a descrição de dado dispositivo. Através da definição e publicação das descrições de dispositivos UPnP, membros do Fórum UPnP podem criar padrões de blocos de construção para uma rede doméstica por exemplo. As normas definidas pelo Fórum UPnP são em plataforma neutra. A adesão e participação na concepção de modelos de esquema de dispositivo estão abertas a todas as companhias membro do Fórum. Empresas interessadas em padronizar classes de dispositivos particulares, são encorajadas a aderir ao Fórum UPnP e participar de comissões de trabalho para criar modelos de esquema para os seus dispositivos.

Fornecedores podem implementar dispositivos que cumprem essas normas, mas elas devem, então, demonstrar que os seus dispositivos tem condições de passar nos testes, a fim de receber a permissão de usar o logotipo em seu dispositivo. O Fórum UPnP fornece os meios para os fornecedores certifiquem seus dispositivos.

O Fórum UPnP também busca promover o padrão UPnP na indústria e com o público em geral. O Fórum fornece uma estrutura para que as empresas possam se reunir e definir normas de bloco de construção: ambas as normas técnicas, como a arquitetura UPnP, e as normas legais, tais como um acordo de desenvolvimento assinado (JERONIMO e WEAST, 2003).

3.2.1 Os comitês do Fórum UPnP

O Fórum UPnP possuem 4 Comitês, organizados de forma permanente.

São os nomes dos Comitês

- Comitê Gestor
- Comitê Técnico
- Comitê de Marketing
- Comissões de Trabalho

3.2.1.1 Comitê Gestor

A Comissão de Coordenação UPnP é o órgão de mais alto nível que direciona o Fórum UPnP. Este comitê possui cerca de 20 membros de várias empresas, incluindo Microsoft. A composição do Comitê Gestor pode mudar ao longo do tempo como a entrada de novos membros. A Comissão de Coordenação fornece liderança empresarial e toma decisões para o Fórum UPnP. Como a equipe de gestão da organização, o Comitê Diretivo supervisiona as comissões de trabalho para definir descrições de dispositivo (DCP). O Comitê Gestor lançou uma empresa separada, a Corporação Implementador de UPnP (UIC), responsável pela certificação de dispositivos.(JERONIMO e WEAST, 2003).

3.2.1.2 Comitê Técnico

O Comitê Técnico da arquitetura UPnP é um grupo de representantes técnicos de várias empresas que processam as questões técnicas de comissões de trabalho. O Comitê Técnico promove as discussões das opiniões, estas questões tem o intuito de produzir os requisitos arquitetônicos. Eles são responsáveis pela "grande figura" tecnicamente falando para o padrão UPnP.(JERONIMO e WEAST, 2003).

3.2.1.3 Comitê de Marketing

O Comitê de Marketing UPnP compromete-se a promoção conjunta dos membros do padrão UPnP, inclusive representando o Fórum UPnP em feiras do setor. Tem a função principal de divulgar a marca e os padrões UPnP.(JERONIMO e WEAST, 2003)

3.2.1.4 Comissões de Trabalho

O trabalho âmago da questão técnica é feito nas Comissões de Trabalho do Fórum UPnP. Estes grupos definem as descrições de dispositivos que descrevem as inter-

faces que o dispositivo fornece para a rede. As comissões de trabalho define a sintaxe e a semântica de um tipo particular de dispositivo de modo que as implementações desse tipo de dispositivo possa a ser intercambiáveis.(JERONIMO e WEAST, 2003).

Para iniciar uma nova comissão de trabalho no Fórum UPnP, os membros devem primeiro fazer uma proposta ao Comitê Gestor UPnP. A proposta consiste de um conjunto de cenários de usuários para demonstrar a utilidade do novo tipo de dispositivo, um cronograma de trabalho proposto, e um compromisso de três grupos independentes para implementar o tipo de dispositivo. Ter múltiplas implementações independentes demonstra interoperabilidade do novo padrão tipo de dispositivo. O grupo é formado com uma carta particular, expressa como um conjunto de objetivos a serem atingidos. Uma vez que o grupo satisfaz a sua carta, seu trabalho está completo e é o grupo se desfaz. Se o grupo decide continuar a trabalhar em uma nova versão da original do dispositivo, o grupo deve ser reagrupado para cumprir os mesmos requisitos que qualquer outro grupo novo para ser organizado.(JERONIMO e WEAST, 2003).

3.3 Componentes de uma rede UPnP

Em uma rede compatível UPnP, existe 3 componentes básicos: dispositivos, serviços e pontos de controle. Os dispositivos podem ser dispositivos controláveis, funcionam como um pequeno servidor, ou seja, disponibiliza serviços, que podem ser acessados de outros dispositivos estes chamados de Pontos de Controle.

3.3.1 Dispositivos UPnP

Um dispositivo pode conter ambas as características, tanto podendo prover serviços como pode conter pontos de controle para invocar serviços de outros dispositivos. Pode ser dispositivos controláveis e também ser um ponto de controle de forma aninhada e os dispositivos que não estão aninhados em outro dispositivo é denominado dispositivo raíz. Todos os dispositivos UPnP disponibilizam um arquivo em XML para fornecer as informações detalhadas para os pontos de controle, tais como fabricante, número serial e dispositivos aninhados.

3.3.2 Serviços UPnP

Os serviços estão disponíveis para realizar uma tarefa. Esse serviço está disponível para os pontos de controle invocarem esta ação. Ela pode retornar um estado do dispositivo ou realizar outra tarefa assim programada, os serviços pode tanto atuar no dispositivo como podem retornar algum valor solicitado. Semelhantes à descrição do dispositivo, essas informações fazem parte de um arquivo XML padronizado pelo Fórum UPnP. O documento de descrição de dispositivo contém um ponteiro (URL) para essas descrições de serviço. Os dispositivos podem conter múltiplos serviços.

3.3.3 Ponto de Controle

O ponto de controle é responsável para procurar os dispositivos e seus serviços disponíveis na rede UPnP, os pontos de controle podem invocar serviços e enviar solicitações para inscrição no servidor de eventos dos dispositivos controláveis, uma vez inscrito no servidor de eventos cada vez que ocorrer o evento o ponto de controle receberá o evento.

3.4 As Fases UPnP

Para os dispositivos compatíveis UPnP, todos devem seguir certos padrões para haver conectividade e compartilhamento entre os dispositivos, estes passo são chamados de fases, cada fase tem uma função definida para no final ter todas as funcionalidades disponíveis.(JERONIMO e WEAST, 2003).

1. **Endereçamento:** Os dispositivos ao se conectar em uma rede devem adquirir um único endereço, para poder se comunicar.
2. **Descobrimto:** O dispositivo procura por um ponto de controle e as descrições de serviços de outros dispositivos.
3. **Descrição:** Os dispositivos fornecem uma lista de todos os próprios serviços disponíveis em um formato padrão.
4. **Controle:** Os dispositivos podem se submeter aos pontos de controle os quais podem enviar comandos para os dispositivos.

5. **Eventos:** Os serviços dos dispositivos notificam mudança de estado para os pontos de controle fazer o registro quando tiver qualquer mudança.
6. **Apresentação:** Os dispositivos UPnP possuem, opcionalmente, uma interface administrativa usando HTML para monitorar e manipular diretamente o dispositivo.

3.4.1 Dependências entre as Fases do UPnP

As fases dentro da arquitetura UPnP, não funcionam de forma independente, ou seja, não funcionam em separado, cada fase só inicia depois que ter terminado uma fase em específico, com exceção da fase de Endereçamento que é a primeira fase do processo, exemplo a fase de Descrição só inicia se a fase de Endereçamento for concluída e assim sucessivamente. (JERONIMO e WEAST, 2003).

A figura 3.1 mostra as dependências de cada fase do processo UPnP, cada fase fornece os passos básicos da Arquitetura UPnP, cada fase tem sua importância no processo, para disponibilizar e compartilhar os recursos oferecidos de dispositivo. (JERONIMO e WEAST, 2003)

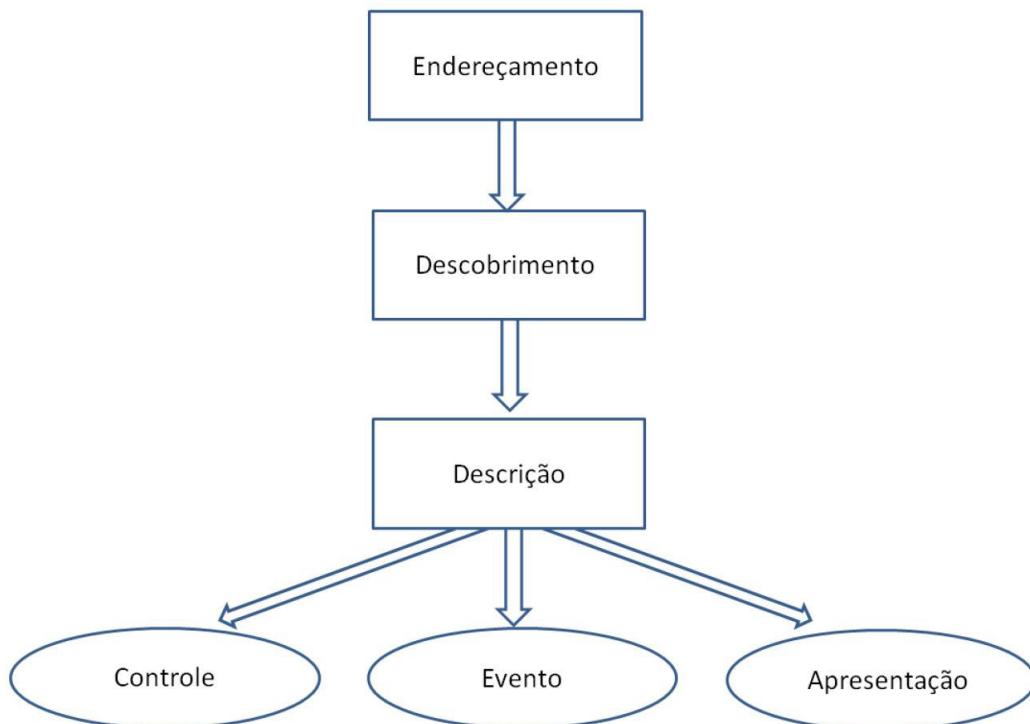


Figura 3.1: Dependências entre as Fases do UPnP (JERONIMO e WEAST,2003).

3.5 A Pilha de Protocolos UPnP

O conjunto de Protocolos TCP/IP e HTTP possibilita a conectividade básica para os dispositivos compatíveis com a arquitetura UPnP. A Comunicação entre os pontos de controle e os dispositivos é feita com o padrão HTTP, como ambos os meios de comunicação é orientado a conexão e protocolos de transporte. TCP é usado para os pontos de controle comunicar-se diretamente com os dispositivos. UDP é usado pelo ponto de controle, caso necessite se comunicar com vários dispositivos simultaneamente (JERONIMO e WEAST, 2003). A base de comunicação entre os pontos de controle e os próprios dispositivos UPnP, é amplamente usado e aprovado pelos usuários da internet em todo o mundo, os desenvolvedores UPnP escolheram esses protocolos por se tratar de uma tecnologia aprovada pelos desenvolvedores da internet e principalmente por ser tecnologia aberta.

3.5.1 Comunicação Ponto a Ponto

As comunicações ponto a ponto tem a finalidade de prover para os pontos de controle, descrição do dispositivo, eventos e pegar a página de apresentação. A figura 3.2 mostra a pilha de protocolos usada para comunicação ponto a ponto usada na arquitetura UPnP.

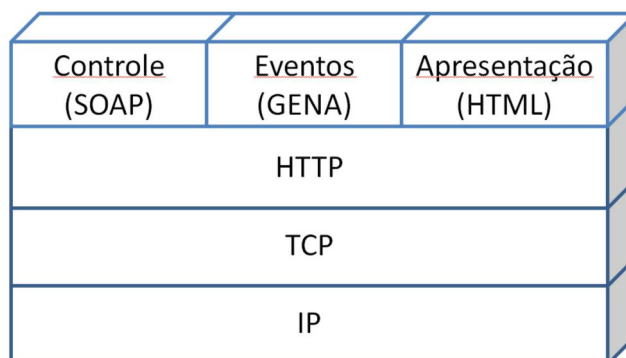


Figura 3.2: Pilha Protocolos Ponto a Ponto (JERONIMO e WEAST,2003)

3.5.2 Comunicação Multicast

Na arquitetura UPnP se faz necessário o uso de comunicação para vários dispositivos de uma só vez, basicamente é usado para anunciar a presença para todos os

dispositivos na rede, justamente para saber quais dispositivos estão disponíveis na rede, na figura 3.3 mostra a dependência da estrutura de protocolos.

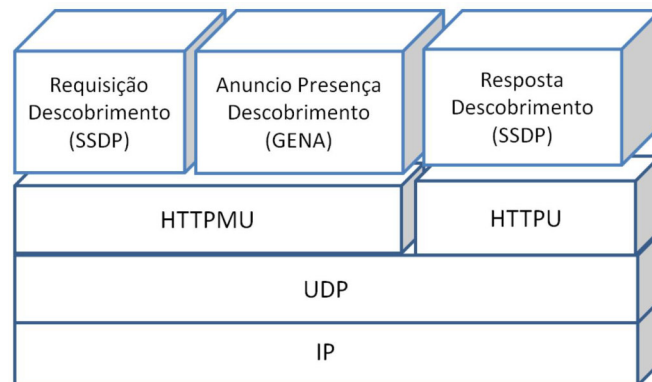


Figura 3.3: Pilha Protocolos Multicast (JERONIMO e WEAST,2003)

4 Pilha de Protocolos UPnP

O estudo dos protocolos da arquitetura UPnP se faz necessário para ter um melhor entendimento do processo envolvido entre os dispositivos e pontos de controle e também no entendimento do funcionamento e em suas limitações envolvidas. Nas seções a seguir vai ser tratado de forma mais aprofundada cada fase e qual protocolo usado em sua implementação.

4.1 Endereçamento dos Dispositivos UPnP

Para haver comunicação entre duas partes, temos que ter um meio para poder enviar mensagens, o modo mais antigo de comunicação a distancia e por meio de carta neste caso à mensagem se encontra dentro de um envelope e fora do envelope existe o endereço do destinatário, e o endereço do remetente, mas para a mensagem chegar ao destinatário o meio de envio e entrega da correspondência precisa saber onde fica o endereço para entregar corretamente a carta isto significa que endereço tem que ser único para o meio de entrega não faça a entrega da carta no lugar errado, o remetente tem que saber o endereço correto tudo isso para que a mensagem chegue ao seu destino.

Na arquitetura UPnP essa é a primeira fase, a fase de endereçamento, onde cada dispositivo precisa de um endereço único para poder integrar a rede de comunicação de dispositivos. O UPnP é construído seguindo os padrões do Protocolo TCP/IP, a qual prove a camada de conectividades entre os dispositivos. Esse representa um passo muito importante, já que se essa etapa não funcionar adequadamente, não vai haver comunicação entre os dispositivos, a rede não terá utilidade nenhuma. No UPnP tem o grande desafio de prove configuração zero em seus dispositivos, logo tem que usar de meios para configuração de endereços IP automaticamente. Existem dois mecanismos para configurar automaticamente os dispositivos DHCP e Auto-IP, tratados a seguir(JERONIMO e WEAST, 2003).

4.1.1 DHCP

DHCP tem finalidade de fornecer de uma forma automática as configurações básicas para os dispositivos terem condições de participar de uma rede de computadores usando o protocolo TCP/IP, nestas configuração básica são passadas; endereço de IP, a mascara de subrede, o endereço do gateway padrão, o endereço do servidor do domínio. O protocolo cliente/servidor e empregado usando o UDP na camada de transporte. O computador ao se conectar em uma rede envia uma mensagem para o Servidor de DHCP pela porta 67, o servidor responde pela porta 68 com as informações básicas de configuração. O servidor de DHCP gerencia os endereços de IP disponibilizando os endereços livres para os novos dispositivos, todos os endereços são armazenados em uma tabela os já utilizados e os disponíveis de acordo com configuração feita pelo Administrador de Rede. O servidor de DHCP centraliza o gerenciamento de IP de uma rede local.

Existem três mecanismos que o Servidor de DHCP usa para determinar endereços para os clientes de rede:

1. Alocação Automática: O servidor de DHCP determina um endereço de IP de forma permanente para o cliente.
2. Alocação Manual: O administrador de Rede determina o endereço para cada dispositivo de rede, o Servidor de DHCP simplesmente disponibiliza o mesmo endereço específico configurado para ele, quando solicitado pelo cliente.
3. Alocação Dinâmica: Este é a configuração mais usadas nos servidores DHCP, e atribuído um endereço para o cliente por um período determinado, quando o cliente deixar a rede ou o tempo expirar, o endereço e disponibilizado para outro cliente.

Cada dispositivo UPnP é construído para ser um cliente de um servidor de DHCP, quando um dispositivo UPnP se conecta em uma rede a sua prioridade e verifica se a rede possui um servido de DHCP para adquirir o seu endereço de rede IP válido, em teoria dispositivo UPnP funciona em qualquer um dos modos de operação do servidor de DHCP, mas o mais eficiente para aplicações com dispositivos UPnP e a alocação dinâmica a qual não requer qualquer configuração do lado do dispositivo UPnP.

4.1.2 AutoIP

Quando é utilizado um servidor de DHCP para atribuir dinamicamente os endereços de IP para os dispositivos, estamos automatizando as configurações básicas, ou seja, não é preciso configurar nada para poder se conectar em uma rede, isto em muito importante porque simplifica bastante o acesso a rede, mas para isso precisamos de um servidor ou um computador rodando sempre, para uma pequena rede, isto se torna muito caro, devido haver necessidade de uma máquina dedicada, manutenção para essa máquina e pessoal especializado para configurar e manter o servidor rodando, tornando essa aplicação em pequenos escritórios mais difícil (JERONIMO e WEAST, 2003). Para manter a premissa de configuração-zero em dispositivos UPnP, foi desenvolvido um mecanismo para se auto configurar na ausência de um Servidor de DHCP na rede local, os próprios dispositivos automaticamente configura o próprio endereço de IP, lógico seguindo algumas regras definidas para o Padrão UPnP, discutidas logo a seguir.

Usando o Auto-IP para determinar os endereços dos dispositivos, a rede se torna acessível aos dispositivos e sem qualquer uso de uma infraestrutura mais complexa como servidor de DHCP e DNS, fazendo com que os dispositivos compatíveis com UPnP, tenham a possibilidade de compartilhar recursos. Habilita a formação de uma rede Ad-Hoc sem qualquer configuração mais complexa, sendo o uso bastante simples.

4.1.2.1 Seleção de endereços

A seleção de endereços no modo de Auto-IP é determinada pelo algoritmo de seleção onde o endereço está contido no intervalo de 169.254/16, sendo estes endereços não roteáveis, ou seja, não passa por roteadores, por segurança os endereços não são acessíveis por redes externas. Após a seleção do endereço o dispositivo atribui a máscara de subrede de classe B 255.255.0.0. A Internet Assigned Numbers Authority (IANA) reservou esta faixa de endereçamento para IP privado, de modo que ninguém poderá usar na Internet. Este intervalo é conhecido como a rede LINKLOCAL. Além disso, o primeiro e o último 256 endereços são reservados para uso futuro e não deve ser selecionado (JERONIMO e WEAST, 2003). No modo Auto-IP, o dispositivo seleciona um endereço em uma faixa pré-determinada, onde a característica principal é que são endereços não roteáveis, os algoritmos dos roteadores de rede não propagam pacotes que o endereço de destino que estão nesta faixa, endereços estes que são usados somente em redes locais, favorecendo a

segurança dos dispositivos, assim não permite que nenhum servidor ou computador ligado a uma rede externa acessar os dispositivos conectados usando os endereços dessa faixa.

4.1.2.2 Resolvendo Conflitos de Endereços

A partir do momento que é escolhido o endereço do dispositivo compatível com UPnP, é necessário verificar se o endereço está sendo usado por outro dispositivo, na configuração usando um servidor de DHCP. O servidor de DHCP possui uma tabela de endereços disponíveis. Mas usando o algoritmo de autoconfiguração é escolhido um endereço de uma faixa pré-determinada e uma verificação deverá ser feita para confirmar se o endereço de IP está sendo usado, para isso é utilizado o protocolo ARP. ARP (Address Resolution Protocol), este protocolo basicamente tem a função de fazer uma correlação entre os endereços de IP e os endereços físicos, para ter comunicação entre os dispositivos, este protocolo é normatizado através do padrão RFC826 o (ARP) Address Resolution Protocol . O ARP permite que um host, um computador por exemplo, encontre o endereço físico de um host destino, tendo apenas o seu endereço IP.

Na arquitetura UPnP o dispositivo ao entrar na rede o mesmo envia um broadcast, uma mensagem de protocolo ARP para todos na rede, usando endereço previamente escolhido, este endereço é escolhido através de um algoritmo, se algum dispositivo responder a pergunta, o endereço está sendo usado, o dispositivo volta a escolher outro endereço e reenvia o broadcast¹, mas se nenhum dispositivo responder, quer dizer que o endereço está disponível e o dispositivo poderá usar o endereço sem problema algum.

4.1.2.3 Limitações do AutoIP

O uso do Auto-IP possui algumas limitações, enquanto que com o uso de DHCP os endereços são configurados por um administrador de rede, esses podem ser roteáveis, ou seja, com uma abrangência bem maior, não fica restrito a uma rede local, neste aspecto tem uma configuração adicional para esta situação. Já na faixa de endereços usado pelo Auto-IP são de uso para rede local, ou seja, não podem ser roteáveis, uma consequência para isso que o ponto de controle de uma rede UPnP tem obrigatoriamente que está na

¹Broadcast ou Radiodifusão é o processo pelo qual se transmite ou difunde determinada informação, tendo como principal característica que a mesma informação está sendo enviada para muitos receptores ao mesmo tempo.

mesma rede local.

4.1.3 Algoritmo de endereçamento dos dispositivos UPnP

Esse é o primeiro passo e o mais importante no processo de comunicação entre os dispositivos, primeiramente o algoritmo verifica se existe um servidor de DHCP na rede, se existir, é solicitado um endereço disponível e termina o algoritmo, após isso o dispositivo atribui o endereço recebido do servidor de DHCP. Caso não tenha um servidor de DHCP na rede o dispositivo entre em modo Auto-IP, neste caso, o dispositivo escolhe um endereço de IP de uma range pré-programado, este endereço é testado se está disponível, para testar se o endereço está disponível é usado o protocolo ARP, onde é feito o Broadcast, ou seja, manda uma mensagem para todos na rede, perguntando se alguém está usando o endereço escolhido, se algum dispositivo responder o endereço está sendo usado, o dispositivo escolhe outro endereço, faz outro Broadcast, até conseguir um endereço disponível, quando algum dispositivo não responder ao Broadcast o endereço está disponível, com o endereço disponível e atribuído o endereço de IP ao próprio dispositivo, finalmente o dispositivo com tempo programado verifica se o servidor de DHCP voltou a funcionar na rede, se o servidor de DHCP voltou a funcionar, e solicitado um endereço e trocado o endereço do dispositivo e o algoritmo termina. Na figura 4.1 mostra o algoritmo de endereçamento dos dispositivos UPnP

4.2 Descoberta dos Dispositivos UPnP

O primeiro passo já foi realizado que é o de se conectar em uma rede e determinar um endereço IP, os dispositivos já possuem conectividade, agora o problema é realizar a interação com os outros dispositivos, os problemas encontrados neste passo são: Como saber se existem outros dispositivos conectados na rede? Como saber quais serviços está disponível na rede? Este e outros problemas são tratados nessa seção chamada de descoberta dos dispositivos UPnP.

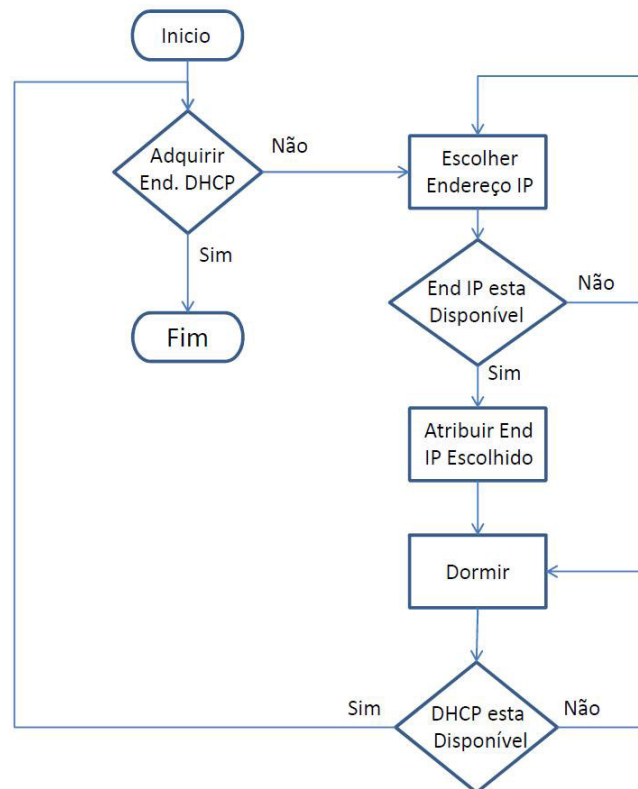


Figura 4.1: Algoritmo de Endereçamento dos Dispositivos UPnP (JERONIMO e WEAST,2003).

4.2.1 Protocolo SSPD

O protocolo Simple Service Discovery Protocol (SSDP) é usado para resolver as questões do descobrimento, o protocolo SSDP é baseado com HTTPU e HTTPMU em uma rede local, este protocolo não requer nenhuma configuração, gerenciamento ou nenhum tipo de administração (JERONIMO e WEAST, 2003). HTTPU e HTTPMU são os protocolos usados para que os pontos de controle localize recursos de interesse na rede e para que os dispositivos anunciem sua disponibilidade na rede. Por meio da definição do uso de solicitações de pesquisa e anúncios de presença. Da mesma maneira, um dispositivo, ao ser conectado à rede, enviará múltiplos anúncios de presença SSDP para promover os serviços que suporta.

4.2.1.1 Identificação de Serviço

O protocolo SSDP introduz dois conceitos relacionados a serviço de identificação, o tipo de serviço e o Unique Service Name (USN). O tipo de serviço é um URI, o

qual identifica o tipo, ou função, ou um recurso em particular, por exemplo, um serviço de impressão. O SSDP não define o tipo de serviço, mas prove uma maneira de ser descoberto.

O Unique Service Name e um URI são usados para identificar de forma única uma instância de um serviço em particular, permitindo que os clientes SSPD tenha como diferenciar entre dois serviços com o mesmo tipo de serviço. O USN normalmente contém um Universal Unique Identifier (UUID) um número com 128-bit usado para identificar de maneira única cada objeto (JERONIMO e WEAST, 2003). Cada UUID é criado por programas utilitários usando o endereço de rede, o horário atual e um componente gerado de forma aleatória, ou seja, mesmo que os serviços tenha o mesmo nome os UUID's sempre vão possuir identificações diferentes para cada objeto.

Exemplo:

service type: "printer"

USN: "uuid:399766A8-B2F5-436d-8B45-0B5BE1F3CA40"

4.2.1.2 Modelo de Comunicação

O modelo de comunicação usado é descentralizado, cada cliente solicita as informações dos recursos disponíveis todos os pontos de controle conectados na rede terão as informações completas sobre o estado de todos os dispositivos enquanto mantém o tráfego da rede baixo, esta é uma das principais vantagens deste modelo de comunicação. O modelo descentralizado é muito robusto e não precisa de um computador central ou qualquer tipo de configuração por um administrador de rede. As informações são sempre disponíveis em tempo real, recebidas diretamente dos recursos disponíveis para os clientes solicitantes (JERONIMO e WEAST, 2003). A única desvantagem é enquanto a rede aumenta o número de dispositivos, também aumenta a largura de banda usada da rede para o processamento do SSDP para resolver o descobrimento e o poder de processamento de cada dispositivo também aumenta, devido cada dispositivo ter que processar as informações de cada recurso disponível na rede se existe vários recursos disponível e cada cliente tem necessidade de escutar e processar cada informação, assim aumenta o processamento do cliente. Neste caso esse modelo é muito eficiente, mas em rede com poucos dispositivos conectados.

4.2.1.3 Requisição de Descoberta e Anuncio de Presença

Existe dois tipos de requisições SSDP a primeira requisição de descoberta, este tipo é usado pelo cliente que solicita os recursos disponíveis na rede. A segunda é o anúncio de presença, permite que os recursos disponíveis ao se conectar na rede permite anunciar sua presença para os clientes. O balanceamento entre as Requisições de Descoberta e o Anuncio de Presença, foi desenvolvida para obter eficiência do protocolo, reduzindo o tráfego de rede, reduzindo o uso da largura de banda da rede ao mínimo, para o uso da descoberta dos dispositivos de rede.

4.3 Descrição dos Dispositivos UPnP

A fase de descrição fica entre a fase de Descoberta e a fase de controle, primeiramente o ponto de controle procura por dispositivos, assim que o dispositivo é anunciado na rede, o próximo passo é saber as características do dispositivo e dos serviços que estão disponíveis, esse passo é chamado de descrição que nada mais é ler um arquivo XML (eXtensible Markup Language) diretamente do dispositivo, nesse arquivo, possui todas as características do dispositivo e as características dos serviços que estão disponíveis. O arquivo XML que descreve o dispositivo, requer um padrão estabelecido pelo Fórum UPnP, ou seja, deve seguir as regras pré-estabelecidas.

O documento de descrição de dispositivo mostra a estrutura lógica do dispositivo, neste documento possui três níveis de elementos: *< specVersion >*, *< URLBase >* e *< device >*. (JERONIMO e WEAST, 2003)

```
<?xml version="1.0"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">

  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <URLBase>base URL for all relative URLs</URLBase>
  <device>
    basic device information elements
    service list
    embedded device list
  </device>

</root>
```

Figura 4.2: Exemplo do inicio um document XML, para um dispositivo

O elemento *< root >* da descrição do dispositivo é a mais alta representação do dispositivo, onde temos um *xmlns* esse tag inclui o esquema padrão onde indica o URI *urn : schemas – upnp – org : device – 1 – 0*. Este TAG é referente ao padrão adotado pelo UPnP Template Language para dispositivos, onde e definido a sintaxes.

O elemento *< specVersion >* contém dois sub-elementos *< major >* e *< mirror >* o qual é setado com 1 e 0, respectivamente significa Versão do UPnP que neste caso é UPnP 1.0. O elemento *< URLBase >* é opcional esse elemento está se referindo a uma URL que define a URL do dispositivo (JERONIMO e WEAST, 2003).

4.4 Controle dos Dispositivos UPnP

Após os dispositivos adquirir um endereço válido, anunciar a presença na rede, os pontos de controle recebe todas as informações dos dispositivos, o ponto de controle tem todas as informações disponíveis para poder invocar ações através dos serviços disponíveis nos dispositivos. Na terminologia UPnP, invocar ações em um dispositivo é chamado de controle. Esta parte está ligada aos pontos de controle em enviar comandos para os dispositivos, nos dispositivos existem serviços que são executados para isso usam-se dois protocolos amplamente usados em computação distribuída são: RPC e SOAP.

4.4.1 Remote Procedure Calls (RPC)

Este protocolo é muito usando em ambientes de programação Cliente-Servidor, computação distribuída, o qual para o programador o uso é simples basta fazer uma chamada de procedimento igual como se fosse uma chamada de procedimento local, mas esta chamada de procedimento é realizada em um computador remoto, a execução é feito em outra máquina, sem qualquer programação adicional, sem qualquer programação de protocolo específica. Temos duas vantagens em usar o protocolo RPC: É muito utilizado e fácil de programação e uso. A desvantagem é que os roteadores normalmente bloqueiam qualquer mensagem RPC, devido o uso muitas vezes está associado a um possível acesso indevido ao computador, tornando uma potencial fonte de ataque ao computador, neste caso é necessário o desbloqueio das portas específicas no roteador ou gateway da rede para o uso adequado do protocolo RPC.

4.4.2 SOAP

O SOAP (Simple Object Access Protocol) Protocolo de Acesso Simplificado a Objetos é um protocolo que usa os protocolos XML e HTTP para enviar comandos baseado em mensagens web em com chamada de procedimento remoto (RPC). XML é usado para representar o conteúdo da mensagem, enquanto HTTP é usado para enviar mensagens para o destinatário. SOAP é especificado de acordo com as convenções, ou seja, tem um formato específico e é formado por quatro partes;

- **Encapsulamento SOAP:** Um esquema XML o qual define o Framework para descrever como é a estrutura da mensagem, como a mensagem deve ser processada.
- **Regras de processamento SOAP:** Outro esquema XML define as regras de processamento das instancias definidas pela aplicação.
- **Ligação SOAP :** Uma conveção usada por diferentes protocolos de transporte. SOAP tem a possibilidade de usar outras combinações com uma variedade de outros protocolos de transporte.
- **Representação RPC em SOAP:** A convenção para representar RPC e as respostas.

As mensagens SOAP são formas básicas para comunicação entre dispositivos. Mensagens SOAP são escritas em XML, fazendo SOAP ser independente de qualquer plataforma, qualquer sistema é capaz de criar e ler arquivos em XML podendo enviar e receber mensagens SOAP. O poder do XML, o protocolo SOAP pode facilmente transmitir e receber estrutura de dados altamente complexos. (JERONIMO e WEAST, 2003). Como não podemos usar o RPC diretamente para rodar procedimentos em computadores remotos, para resolver esse problema foi criado o protocolo SOAP, que nada mais é encapsular uma mensagem RPC, para poder executar procedimentos remotos nos dispositivos. Então temos uma chamada RPC encapsulada em XML, onde XML é usada para descrever como deve ser executada a chamada e depois encapsulado em HTTP o qual é transmitida para o dispositivo, HTTP na camada de transporte de rede. Assim que a mensagem de controle e recebido pelo dispositivo e feito o processo inverso, ate encontrar a chamada RPC e assim executar.

4.5 Eventos em Dispositivos UPnP

Temos até agora um dispositivo, interligado na rede, descoberto e o ponto de controle tem todas as informações sobre os serviços disponíveis no dispositivo. O ponto de controle tem a possibilidade de monitorar variáveis do dispositivo, podemos chamar de evento todas as vezes que essa variável tiver alguma mudança em seu valor, isso é chamado de evento, que nada mais é que uma mensagem enviada pelo dispositivo para o ponto de controle com o novo valor da variável, para isso o ponto de controle tem que registrar o novo evento, esse evento tem que está disponível no dispositivo. O protocolo usado para esta funcionalidade é o protocolo General Event Notification Architecture (GENA). Esta funcionalidade é muito importante dentro da arquitetura UPnP, para poder monitorar o estado dos dispositivos de uma maneira simplificada, receber os estados dos dispositivos de interesse. De uma maneira bem simples os pontos de controle estarão sempre com o último estado dos dispositivos monitorados, sem requerer qualquer atualização de estado, ou seja, não precisa mandar uma mensagem para receber o seu estado, simplesmente receber a mensagem automaticamente quando a mudança de estado ocorrer no dispositivo.

4.5.1 Eventos em um sistema Distribuído

Os componentes para comunicação em um sistema distribuído geralmente usa dois mecanismos: RPC Remote Procedure Call e notificação de eventos. As notificações de eventos é o menos utilizado dos dois modelos. No modelo RPC, os objetos estão esperando passivamente os clientes através dos serviços, enquanto que notificação de eventos é baseado em mudança de eventos, os sistemas são modelados a eventos, permitindo que outros objetos para responder dinamicamente, talvez invocando serviços prestados por ainda outros objetos (JERONIMO e WEAST, 2003). São modelos amplamente utilizados no caso do UPnP, ambos são usados para prover funcionalidades distintas, um para executar serviços remotos e o outro para prover notificação eventos, de uma maneira transparente e simples.

4.5.2 General Event Notification Architecture (GENA)

O protocolo GENA funciona como analogia de um sistema publicador e assinante, sendo na arquitetura UPnP, o publicador é representado pelo Dispositivo e o

Assinante e representado pelo Ponto de Controle. O assinante pode fazer uma solicitação, renovação ou cancelamento de uma assinatura de eventos. Primeiro o assinante envia uma mensagem de inscrição para o publicador, se a assinatura for aceita o publicador envia um ID da inscrição e a validade de sua inscrição. Para as outras operações sobre a assinatura deve ser usado o ID da inscrição, como para renovação e cancelamento da assinatura, este ID é referente a sua assinatura em específico. Antes do vencimento da Assinatura o Assinante deve renovar a assinatura ou o publicador pode cancelar a assinatura, assim como se o assinante não estiver mais interessado na assinatura pode ser feito o cancelamento para o publicador.

Seguindo o padrão de protocolos aberto, bastante usado e difundido, os quais são usado pelo UPnP, O protocolo GENA usa a notificação de eventos pela web. GENA Usa o protocolo HTTP na camada de transporte para comunicação entre os publicadores e os Assinantes, a figura 4.3 mostra a pilha de protocolos usado pelo GENA.(JERONIMO e WEAST, 2003).

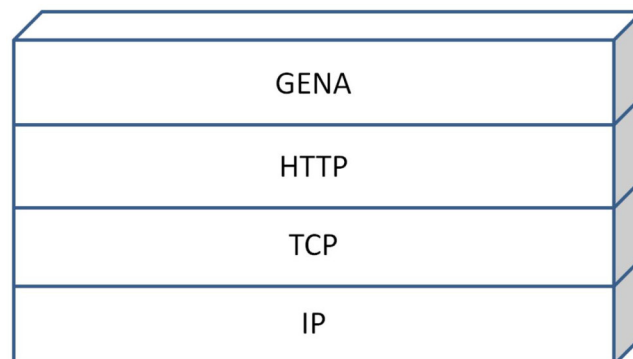


Figura 4.3: Pilha de Protocolos GENA

4.5.2.1 Métodos e cabeçalhos em HTTP usados pelo GENA

GENA apresenta três métodos HTTP que são usados para gerenciar inscrições em eventos e entregar mensagens:

- *SUBSCRIBE* Para se inscrever para receber notificações de eventos e renovar uma subscrição existente. Os cabeçalhos será diferente dependendo de qual função se destina.
- *UNSUBSCRIBE* Para cancelar uma assinatura.

- *NOTIFY* Para enviar uma notificação de evento para um assinante. GENA apresenta os seguintes cabeçalhos que são usados com os novos métodos de HTTP:
- *CALLBACK* É usado para comunicar uma URL a ser usada para chamar de volta a outra entidade. Por exemplo, o assinante envia um cabeçalho de retorno de chamada quando se registrar para receber *publisherevents*. É usado essa URL ao enviar notificações de eventos.
- *NT* É o tipo de notificação. Ele é usado para informar o assinante que tipo de notificação que é isso.
- *NTS* É o tipo sub-notificação. Ele permite maior refinamento do tipo de notificação.
- *SID* É o ID (identificador) de subscrição. Este ID é gerado pelo publicador para fazer referência a uma assinatura. Tanto o publicador do assinante e usar esse ID quando se comunicar com o outro e se referindo a uma subscrição particular.

Mensagens GENA também usar os cabeçalhos HTTP padrão como Host, Timeout, Data Server, Content-Length, e Content-Type.(JERONIMO e WEAST, 2003).

4.5.2.2 Usando o GENA em Aplicação UPnP

Em termos de conceito o protocolo GENA é uma implementação de publicador e assinante, facilmente encontrado em aplicações usando UPnP, no ponto de vista UPnP os pontos de controle são os assinantes e os serviços UPnP são os publicadores. Os dispositivos não são os publicadores de eventos, como poderia se imaginar, mas existe uma implementação de um serviço exclusivo dentro dos dispositivos para realizar a tarefa de publicador, podemos chama-lo de serviço de notificação (JERONIMO e WEAST, 2003).

O protocolo GENA é de simples implementação e largamente usado em sistemas computacionais distribuído, implementa o conceito de publicador e assinante, tem uma grande característica que é a de simplificar o tráfego de mensagens na rede, trafegando somente as mensagens com as informações importantes para o assinante. De uma maneira bem sucinta, o assinante possui a lista das variáveis disponíveis para ser monitorada e escolhida a variável é enviado uma mensagens para o publicador, neste caso o dispositivo, que retorna se a assinatura foi aceita, se a assinatura foi aceita, o assinante, neste caso o ponto de controle, recebe uma mensagem com um número da assinatura e

uma validade, a partir deste ponto, o assinante receberá uma mensagem todas as vezes que a variável monitorada sofre uma mudança, todos os assinantes desta variável, receberá a mensagem com o novo valor. O assinante tem que renovar a assinatura, se o convém, antes do vencimento, se não deixará de receber as mensagens referente a solicitação de assinatura inicial.

A arquitetura UPnP estabelece convenções adicionais para eventos, além dos já especificados pelo protocolo GENA .

4.5.2.2.1 Descrição de Serviços e Variáveis de Estados Monitoráveis: No serviço de Descrição UPnP foi visto que o dispositivo retorna ao ponto de controle uma lista de serviços disponíveis e uma lista de variáveis de estado do próprio dispositivo, essas mesmas variáveis podem ser monitoradas, uma ou mais variáveis podem ser monitoradas pelo ponto de controle, bastando realizar uma requisição para o dispositivo enviando à lista das variáveis que vão ser monitoradas, feita a requisição, o dispositivo retorna o número ID da assinatura feita e sempre que houver qualquer mudança na variável o serviço de notificação do dispositivo vai enviar uma mensagem com o novo valor da variável, enquanto que a assinatura seja válida, em tempos e tempos o ponto de controle deve enviar uma mensagem revalidando a assinatura referente as variáveis monitoradas

4.5.2.2.2 Mensagem Baseadas em XML: Os serviços enviam mensagens para os pontos de controle indicando mudança das variáveis monitoradas nos dispositivos, estas mensagens contêm os nomes de uma ou mais variáveis e os valores de cada variável, todas usando a sintaxe XML, de acordo com o padrão estabelecido pelo Fórum UPnP.

4.5.2.2.3 Mensagem Inicial de Monitoração: Esta mensagem especial é enviada pelo ponto de controle quando ao dispositivo quanto à realização de uma assinatura, nessa mensagem contêm o nome das variáveis as quais necessita da monitoração pelo ponto de controle. É a primeira mensagem do modelo de Monitoração.

4.6 Apresentação

Dentro da arquitetura UPnP, temos os componentes participantes da rede que são os dispositivos e os pontos de controle, estes pode interagir de forma automática sem

intervenção humana, visto anteriormente, mas umas das funcionalidades que todos os dispositivos possuem é a apresentação onde se pode interagir com um dispositivo usando uma interface HTTP, uma página web, ou seja, os dispositivos possuem um pequeno servidor de página HTTP, esta página pode conter informações do dispositivo e comandos, disponibilizados para ter acesso aos estados e interagir com o dispositivo, sem qualquer ferramenta adicional, necessitando somente de um programa browser comum, um interpretador HTTP simples, podendo ser até um dispositivo pessoal com menos poder de processamento um celular por exemplo. A página de apresentação não é uma parte essencial da arquitetura, pode ser encontrado ou não nos dispositivos, esta funcionalidade é opcional, todos os dispositivos, por padrão UPnP, possuem esta funcionalidade, mas precisam ser ativada e programada em cada dispositivo pelo programador do dispositivo.

Em resumo a fase de apresentação é muito simples bastando lembrar alguns pontos importantes:

- Todos os dispositivos UPnP podem prover um acesso a interface administrativa web para monitorar e controlar através da página de apresentação do dispositivo. Podemos associar esta funcionalidade com as encontradas em roteadores residenciais onde a configuração é feita por uma página web, muito prático e não necessita de instalação de programas específicos.
- A página de apresentação é um recurso opcional. Os desenvolvedores dos dispositivos tem a liberdade para determinar o design das páginas web o conteúdo e os recursos usados em suas páginas web, em seus dispositivos.
- Os desenvolvedores podem escolher como a implementação deve ser feita tendo a liberdade de aumentar a complexidade tanto internamente no dispositivo com fora do dispositivo, dependendo muito do poder de processamento interno dos dispositivos com dos pontos de controle.
- Localização de páginas de apresentação se baseia em mecanismos existentes presentes em HTTP e HTML para especificar o idioma para o usuário e a codificação de conteúdo.

5 Cenários de Uso

Neste capítulo é exemplificado como construir um dispositivo simples através das ferramentas de desenvolvimento da Intel, como testar o dispositivo recém-criado, mostrar um exemplo de aplicação com Áudio e Vídeo na arquitetura UPnP, mostrar um exemplo de servidor de mídia em código aberto e estudo do padrão DLNA para equipamentos de reprodução de Áudio e Vídeo.

5.1 Desenvolvimento de Dispositivos UPnP

Para desenvolver um dispositivo compatível com UPnP é necessário fazer a especificação física, e depois a sua especificação lógica e configurar as suas variáveis de estado.

Quando se usa o SDK Intel para UPnP para gerar os dispositivos de rede, ele dispensa o desenvolvedor de lidar com questões complexas, como instalação, configuração, manutenção, protocolos e software. Todas estas questões são tratadas pelo SDK e as tarefas associadas e processos são transparentes para o desenvolvedor. Essa tecnologia permite que pessoas sem o conhecimento básico de redes, possa facilmente instalar, utilizar, atualizar e manter os produtos cada vez mais sofisticados. Mais informações sobre Intel SDK UPnP no Anexo B.

Antes, projetar e implementar um dispositivo UPnP era necessário um investimento considerável de tempo na geração de arquivos, descrição de serviço, construção de código-fonte, teste e depuração. Não existia ferramentas simples e úteis para automatizar qualquer um dos processos. O processo de teste era especialmente trabalhoso, e a depuração ineficaz, logo, o produto final não tinha uma boa qualidade nem compatibilidade com outros dispositivos.(INTEL,2002).

As fases de desenvolvimento usando o Intel SDK UPnP são mostradas na figura 5.1.

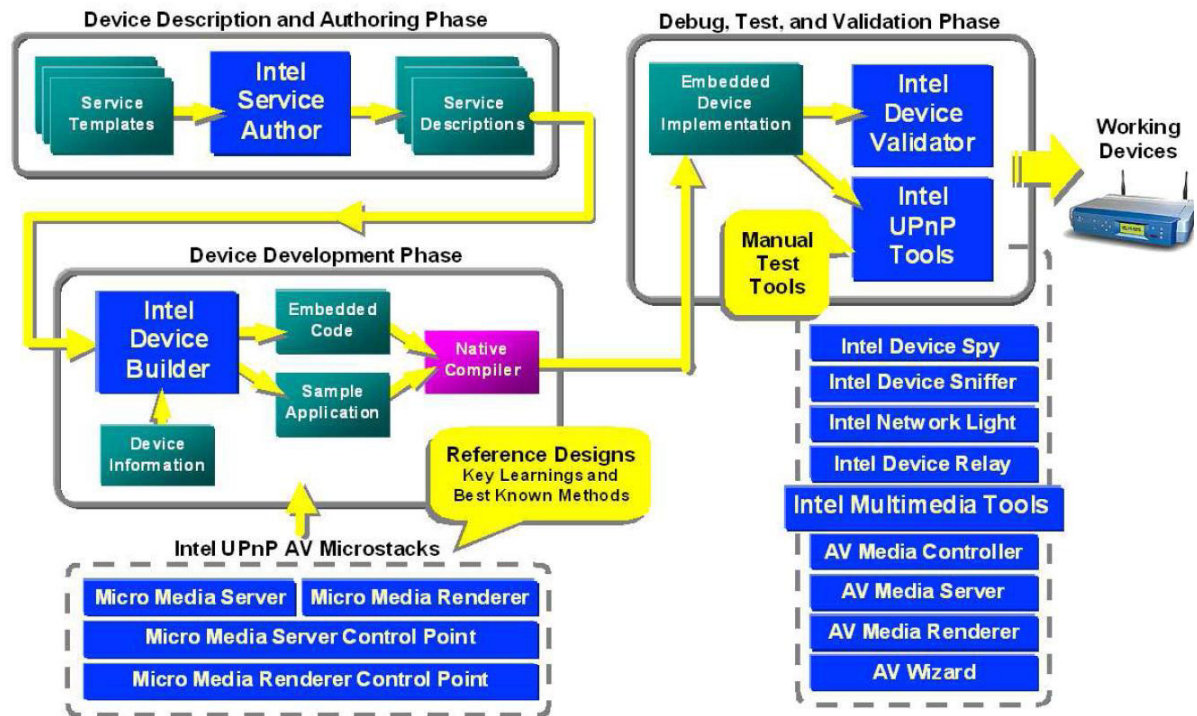


Figura 5.1: Fases desenvolvimento usando Intel SDK para UPnP (INTEL,2002)

5.1.0.3 Criação descritiva dos serviço utilizando Intel Service Autor

Descrição dos serviços é o processo de criação dos serviços e os parâmetros, ou seja, criar o modelo de serviço. Um dispositivo UPnP tem um ou mais descrições de serviços, o qual mostra as capacidades do dispositivo.(INTEL,2002). Esse passo é o primeiro e mais importante onde vai ser definido o alicerce do dispositivo, os serviços e as variáveis de estado passíveis de monitoração, ou seja, quais serviços vão ser disponibilizados e quais variáveis de estado o dispositivo vai possuir.

5.1.0.4 Desenvolvimento de dispositivo Usando Intel Device Builder

Device Builder recebe os arquivos .xml e gera o código embutido de um aplicativo de teste. Isso libera os desenvolvedores da tediosa tarefa de criar lógica da pilha de protocolos UPnP e permite que eles se concentrem no desenvolvimento do dispositivo a nível de lógica. Device Builder também prevê a criação de dispositivos UPnP com conteúdo para dispositivos embutidos UPnP. (INTEL,2002).

5.1.0.5 Depuração e Teste de utilização de dispositivo usando o Intel Device Spy

Essa ferramenta tem a finalidade de testar o dispositivo recém criado para verificar a compatibilidade perante o padrão UPnP.(INTEL,2002).

5.2 Exemplo Simples de Criação de um Dispositivo UPnP: LampadaIP

Abaixo os passos para gerar um código de um dispositivo UPnP usando a Intel SDK para UPnP. O dispositivo vai se chamar LampadaIP vai ter um serviço chamado de SwitchPower e uma variável de estado "status". A criação desse dispositivo é para ilustrar o uso das ferramentas SDK da Intel para UPnP.

5.2.1 Primeiro passo: Gerar o arquivo de Especificação do Dispositivo

Para iniciar é necessário o arquivo de especificação dos serviços do novo dispositivo. É usado o ferramenta chamada Service Author. No dispositivo vai existir uma variável de estado e um serviço, verificamos na aba **State Variable**, mostrado na figura 5.2.

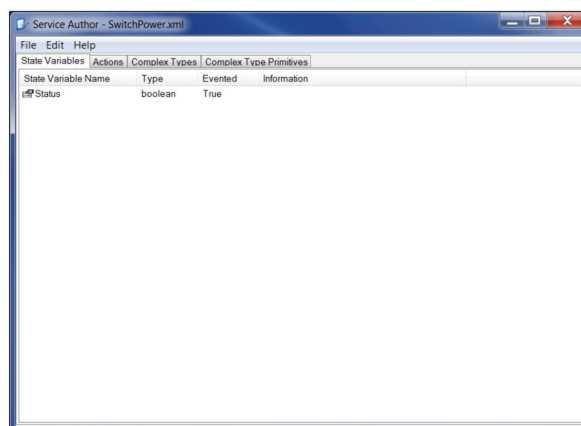


Figura 5.2: SDK Service Author aba State Variable

E na aba **Action** é verificado o serviço criado, mostrado na figura 5.3.

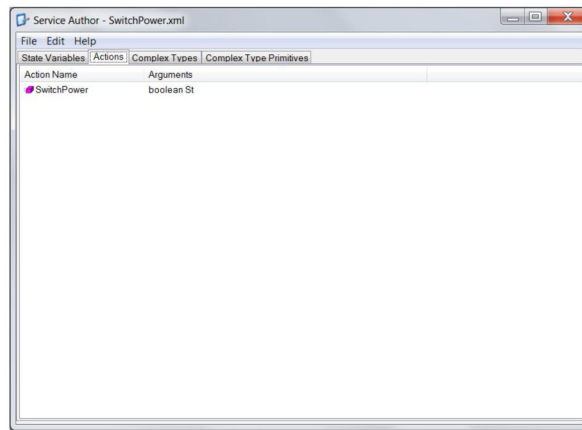


Figura 5.3: SDK Service Author aba Action

O Service Author gera um arquivo .xml que vai se chamar de SwitchPower.xml mostrado na figura 5.4

```
<?xml version="1.0" encoding="UTF-8"?>
- <scpd xmlns="urn:schemas-upnp-org:service-1-0">
  - <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  - <actionList>
    - <action>
      <name>SwitchPower</name>
      - <argumentList>
        - <argument>
          <name>St</name>
          <direction>in</direction>
          <relatedStateVariable>Status</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
  </actionList>
  - <serviceStateTable>
    - <stateVariable sendEvents="yes">
      <name>Status</name>
      <dataType>boolean</dataType>
    </stateVariable>
  </serviceStateTable>
</scpd>
```

Figura 5.4: SDK Service Author SwitchPower.xml

O arquivo SwitchPower tem todas as informações sobre o serviço disponibilizado pelo novo dispositivo LampadaIP.

5.2.2 Segundo passo: Gerar e Compilar o código em C#

Com o arquivo das especificações dos serviços SwitchPower pronto, pode-se criar o código do dispositivo, usando a ferramenta chamada "Device Builder".

Para criar um dispositivo basta clicar em **new device** na janela a esquerda e colocar o nome do dispositivo LampadaIP, conforme a figura 5.5

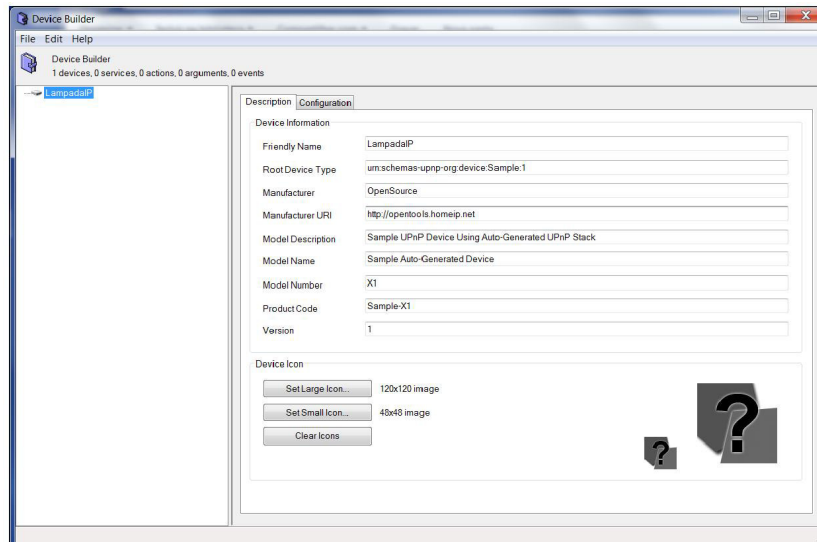


Figura 5.5: SDK Device Builder LampadaIP

O dispositivo foi criado agora precisa inserir os serviços que escrevemos no passo anterior, no dispositivo clicar em **add service from file...** e seleciona o arquivo criado no passo anterior o SwitchPower.xml, a tela deve ficar conforme a figura 5.6.

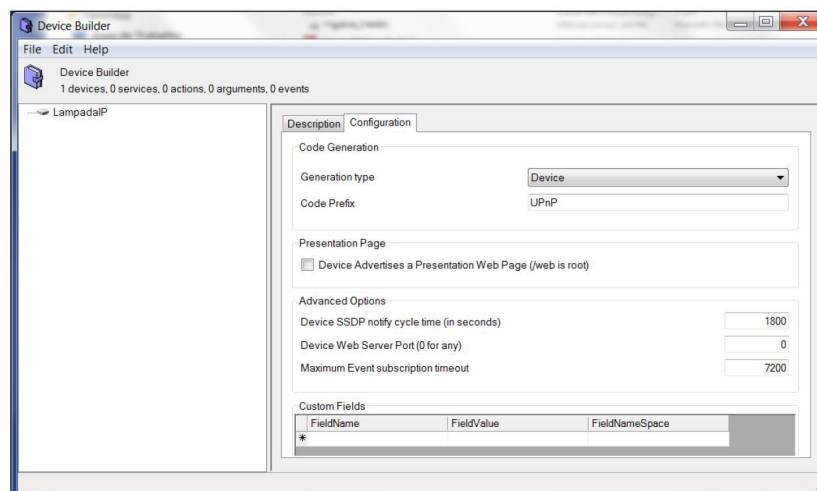


Figura 5.6: SDK Device Builder Adicionar Serviço

Na aba **Configuration**, tem que selecionar **Device** na opção **Generation Type**, o Device Build está pronto para gerar o código do dispositivo, a tela deve ficar conforme a figura 5.6

Agora basta gerar o código no Menu **File** selecionar **Generate Stack**, escolher a plataforma, neste caso e .NET Framework Stack (C#) e a pasta onde será gerado os arquivos, a tela deverá ficar conforme a figura 5.7.

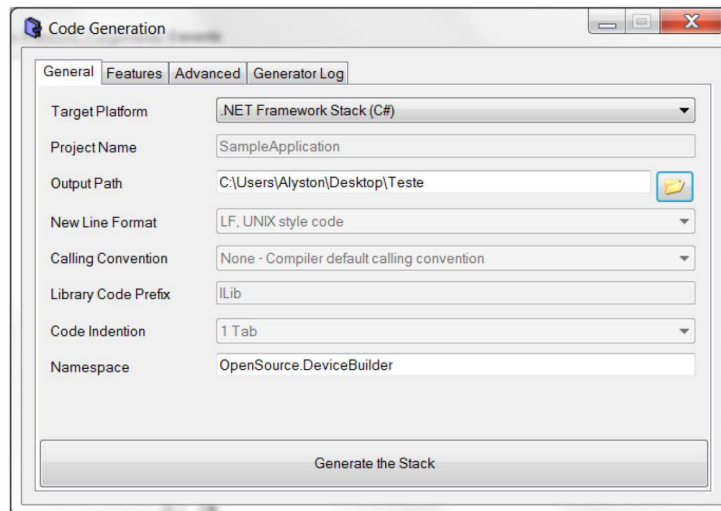


Figura 5.7: SDK Device Builder Generate Stack

Depois de configurado a tela **Generate Stack** corretamente, basta apertar o botão **Generate Stack**, e verificar os arquivos que o Device Build gerou e compilar os arquivos e executar o programa do dispositivo LampadaIP. O arquivo fonte completo está no anexo A.

O Aplicativo LampadaIP possui somente a pilha de protocolos UPnP, ele vai rodar usando a saída padrão do Windows.

5.2.3 Terceiro passo: Testar o dispositivo recém criado.

Depois de executado o programa do dispositivo LampadaIP, gerado no passo anterior é necessário fazer o teste, ao executar o Device Spy, que nada mais é um ponto de controle universal, para visualizar a rede UPnP para verificar quais são os dispositivos UPnP conectados na rede. Verifica-se se existe algum dispositivo chamado LampadaIP e se o dispositivo estiver funcionando corretamente, vai aparecer na árvore de dispositivos, conforme a figura 5.8

Para testar o dispositivo LampadaIP, vamos invocar o serviço criado anteriormente no primeiro passo, o SwitchPower, passamos um argumento binário, verdadeiro ou falso, simulado o estado da Lâmpada. Quando selecionamos "True" e clicamos no botão **Invoke** é enviado o comando para o dispositivo e ele imprime na saída padrão o comando **ImportedService.SwitchPower(True)**, repetindo o comando passando "False", temos a impressão na tela de saída padrão do sistema operacional o evento e o argumento

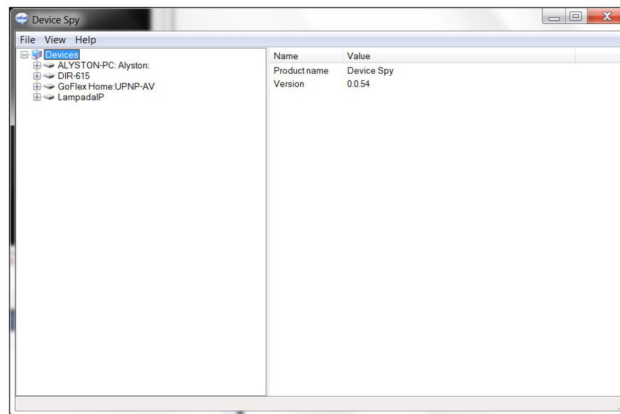


Figura 5.8: SDK Device Spy Teste da LampadaIP

passado `ImportedService_SwitchPower(False)`. Logo verificamos que o dispositivo LampadaIP está funcionando corretamente. Simulação mostrada na figura 5.9.

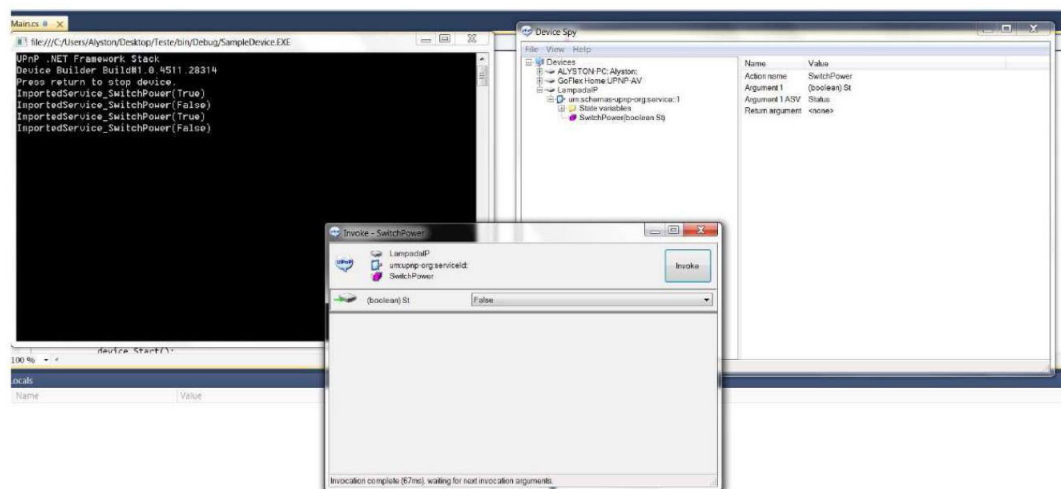


Figura 5.9: SDK Device Spy Teste Serviço da LampadaIP

5.3 Arquitetura UPnP em Áudio/Vídeo

Seguindo o modelo da arquitetura UPnP, nesta deve existir um ponto de controle, interagindo com um ou mais dispositivos, No caso da arquitetura UPnP A/V deve existir pelo menos dois dispositivos, um deles chamada do Media Server e o outro Media Renderer. Na figura 5.10 mostra arquitetura mínima para UPnP A/V. (JERONIMO e WEAST, 2003).

A definição dos dois dispositivos: Media Server, tem a função de armazenar

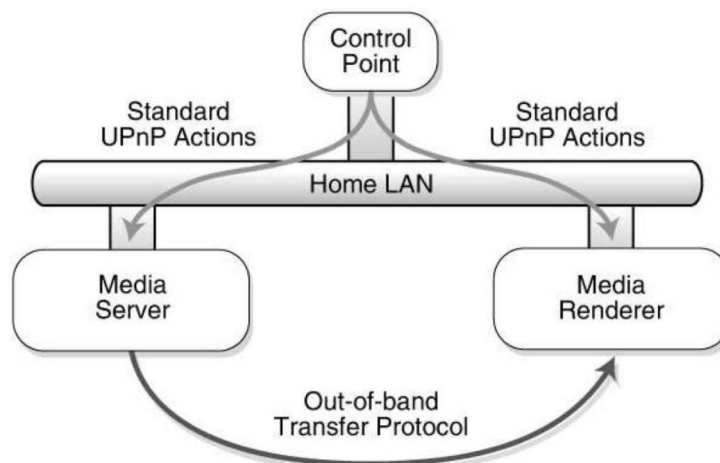


Figura 5.10: Arquitetura mínima para UPnP A/V (JERONIMO e WEAST,2003)

o conteúdo, ou seja, os arquivos, já o Media Renderer tem a função de reproduzir os arquivos armazenados no Media Renderer, o ponto de controle tem a função de juntar os dois dispositivos e comandar a reprodução da Media. (JERONIMO e WEAST, 2003).

A especificação padrão para a arquitetura UPnP A/V esta disponível em <http://www.upnp.org>, são elas:

- Arquitetura UPnP AV
- Modelo Dispositivo Media Server
- Modelo Dispositivo Media Renderer
- Modelo Serviço de Controle Renderer
- Modelo Serviço Gerenciamento de Conexão
- Modelo Serviço Transporte AV
- Modelo Serviço Diretório de Conteúdo.

Cada especificação define os requerimentos básicos, dos estados das variáveis e ações dos serviços, essas especificações garantem a compatibilidade entre os equipamentos.

5.3.1 Cenário de Uso: Reprodução de Áudio

Para reproduzir um arquivo é necessários ter os recursos disponíveis na rede, o usuário interage com os dispositivos através do ponto de controle, o ponto de controle

interage com os outros dois dispositivos, o Media Server e o Media Rederer, formando o sistema mínimo para reproduzir os arquivos de mídia. A interação entre os dispositivos e ponto de controle, se faz conforme o diagrama de Seqüência de mensagem , mostrado na figura 5.11, mostrando cada mensagem, para obter no final o comando para a reprodução do arquivo escolhido.

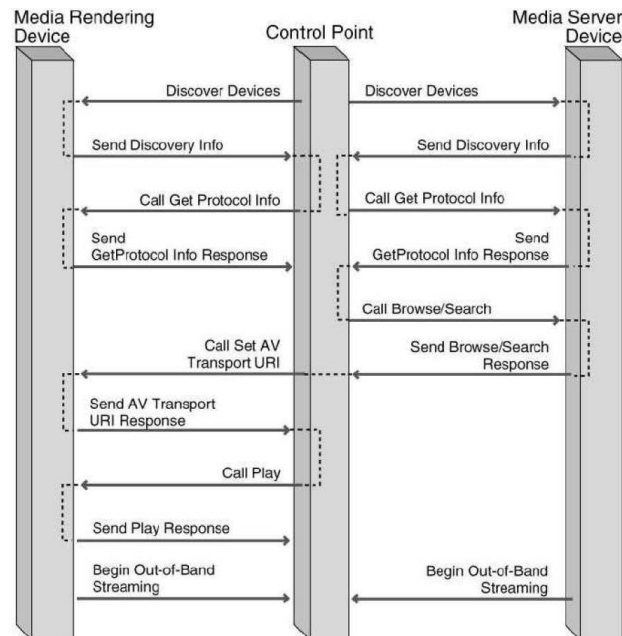


Figura 5.11: Diagrama de Sequencia Mensagens UPNP A/V (JERONIMO e WEST,2003)

Este diagrama da figura 5.11 ilustra a quantidade de mensagens que são trocadas entre os dispositivos e ponto de controle para iniciar a uma simples reprodução de um arquivo de mídia. Verifica-se o alto uso da rede, se a transferência de mídia se fizer pelo mesmo meio físico, poderá haver um atraso na transferência do arquivo ou no envio e retorno de comandos. Verifica-se que a ausência de configuração causa um uso acentuado dos recursos de rede, não impacta em problema até um certo ponto, onde o número de dispositivos pode consumir toda a largura de banda disponível no meio físico de rede.

5.4 Aplicação em Automação Residencial

Um dos principais objetivos do Padrão UPnP é a automação residencial, para facilitar o uso dos usuários domestico, bastando ter que ligar e usar, pode-se até ilustrar um exemplo de uso residencial

Chegando em casa depois de um longo dia de trabalho, decide assistir a um filme. Ele passa pela na cozinha, olhar pela tela de cristal liquido da geladeira e verifica os filmes disponíveis, em seu servidor de mídia, seleciona o filme para assistir. O programa de seleção do filme comanda todo o sistema de home theater e inicia automaticamente o filme. O programa central que controla o ambiente também escurece as luzes e ajusta o volume dos alto-falantes.

Esse cenário já é possível, se a televisão é compatível com UPnP AV Media Renderer. Hoje um computador pode ser o servidor de midia, junto com o controlador centralizado que vai ajustar o ambiente, incluindo lâmpadas.

Em um ponto controle central seria armazenado as preferenciais dos eventos, para cada ambiente e faria o acesso aos dispositivos para enviar os comandos. O celular poderia ser a fonte dos comandos, via um navegador acessaria a página de apresentação do controle central.

Existe outras possibilidades de cenários, possíveis de serem implementados

5.5 Servidor de Mídia UPnP - Media Tomb

MediaTomb é de fonte aberta sob licença GPL, usa o padrão UPnP Media Server com uma interface web muito agradável e simples, ele permite que você possa transmitir suas mídias digitais através de sua rede doméstica e ou rodar em uma grande variedade de dispositivos compatíveis com UPnP. MediaTomb implementa o Padrão UPnP MediaServer Versão 1.0. (BATSYAN, BOSTANDZHAYAN, WIMMER,2012).

A implementação atual concentra-se em componentes que são exigidos pela especificação, no entanto para estender a funcionalidade para cobrir as partes opcionais da especificação. MediaTomb deve funcionar com qualquer MediaRenderer compatível com UPnP.(BATSYAN, BOSTANDZHAYAN, WIMMER,2012).

O Media Tomb tem um servidor para navegar no sistema integrado de arquivos acesso pela página de apresentação, o que significa que qualquer pessoa que tenha acesso à interface do usuário pode navegar pelo seu sistema de arquivo (com permissões de usuário sob o qual o servidor está em execução) e também baixar os dados onde o servidor está rodando.(BATSYAN, BOSTANDZHAYAN, WIMMER,2012).

O MediaTomb tem como característica disponibilizar os arquivos em uma Rede UPnP, logo, é muito fácil disponibilizar os arquivos para uma rede, assim é recomendado, somente usar em redes locais, com no mínimo uma segurança ativa, firewall ou wifi criptografado, para não permitir acesso indevido aos arquivos do computador disponibilizados pelo MediaTomb.

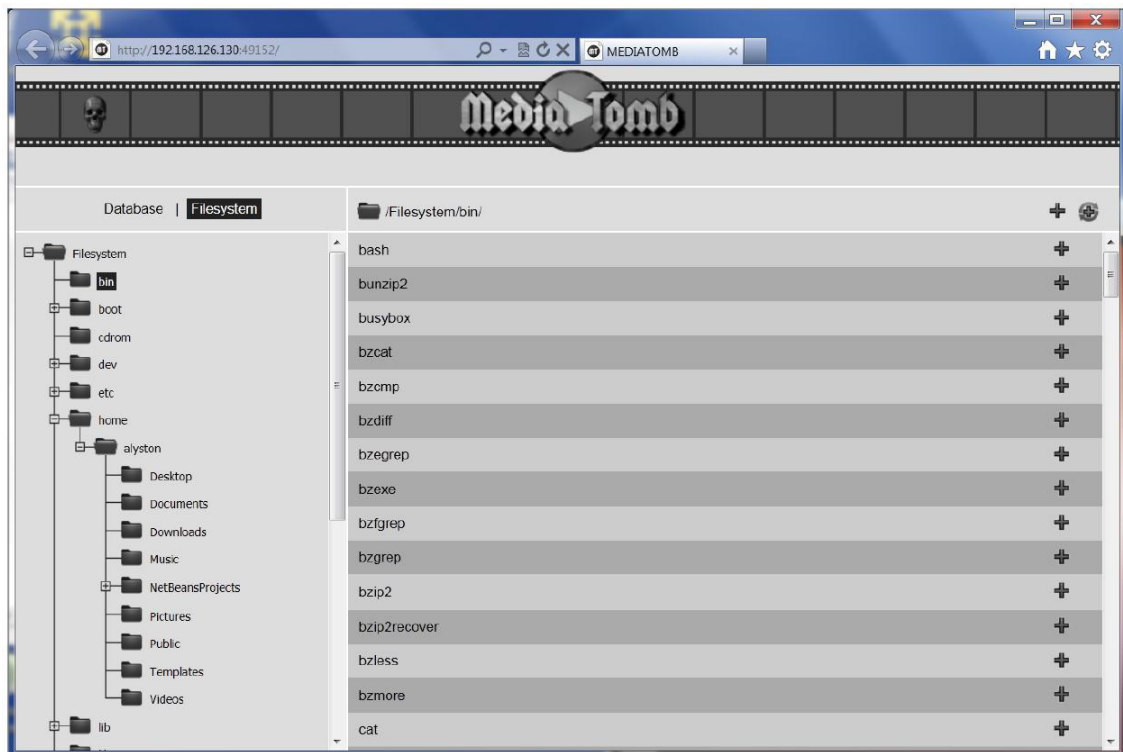


Figura 5.12: Página de Apresentação do MediaTomb

5.5.1 Recursos Suportados pelo Servidor de Mídia

Estes são alguns dos recursos suportados pelo servidor de Media UPnP Media Tomb.

1. Navegar e reproduzir mídias via UPnP
2. Extração de arquivos metadados do mp3, ogg, flac, jpeg, etc .
3. Página de apresentação sofisticada, com uma exibição de árvore dos dados e do sistema de arquivos, permitindo adicionar, remover, editar e procurar.
4. Suporte para URLs externos (criar links para conteúdos da internet e servi-los via UPnP para o seu processador)

5. Configuração altamente flexível, permitindo-lhe controlar o comportamento de vários recursos do servidor de Mídia.
6. Suporte para Linux, FreeBSD, NetBSD e Mac OS X
7. Funciona em x86, Alpha, ARM, MIPS, Sparc e PowerPC2.

5.5.2 Teste da Página de Apresentação Media Tomb

Como teste, na mesma máquina física, rodando Windows 7 Ultimate 64 bits, foi instalado uma máquina virtual com Ubuntu 12.04 LTS, nesta máquina virtual foi instalado o servidor de Mídia Media Tomb. Foi habilitado no Media Tomb a página de apresentação do servidor de mídia. No Windows 7, com o programa Device Spy da Intel SDK UPnP, foi encontrado o dispositivo chamado Media Tomb e na página de apresentação é verificado o endereço da página de apresentação, neste caso foi o `http://192.168.126.130:49152/`, conforme visto na figura 5.12 que é o acesso ao servidor via o Internet Explorer do Windows, vale resaltar que não foi pedido nenhuma senha e todos os arquivos foram disponibilizados em um programa navegador, fora do sistema operacional do servidor.

Ao usar este tipo de aplicativo, evidente a facilidade de compartilhamento dos arquivos do sistema operacional, compatíveis com a Arquitetura UPnP. O cuidado de não disponibilizar arquivos importantes para a rede em que está conectado.

5.6 DLNA - Digital Living Network Alliance

DLNA é uma aliança de fabricantes, formada em 2003, atualmente com mais de 240 empresas de diversos setores da área de tecnologia. Dentre essa 26 empresas são responsáveis por desenvolver essa nova tecnologia. As empresas responsáveis pelo desenvolvimento são: Sony, Intel, Microsoft, Motorola e muitas outras. O conselho é formado pelas empresas, AwoX, Broadcom, Intel, Microsoft, Nokia, Panasonic, Samsung, Sony e Technicolor, elas são responsáveis por supervisionar o trabalho de outras quatro comissões responsáveis por cada parte: Marketing, Técnica, Ecosistema e Certificações. (SONY,2012).

O grande desafio do DLNA é alcançar a interoperabilidade entre diversos fabricantes e desenvolver dispositivos capazes de fornecer aos usuários meios para facilitar o compartilhamento multimídia em um ambiente doméstico utilizando principalmente a tecnologia sem fios Wireless.(DLNA,2012)

5.6.1 Característica DLNA

O DLNA funciona com até três tipos de conexões, Wireless, Ethernet e MoCA¹, na figura 5.13 as principais características técnicas do DLNA junto com a pilha de protocolos. (SONY,2012).

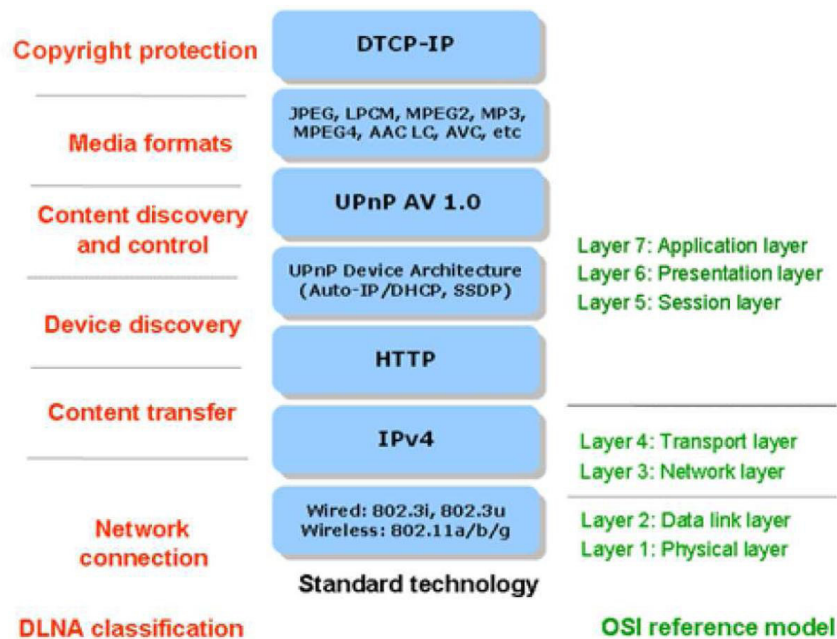


Figura 5.13: Pilha de Protocolos DLNA (SONY,2012)

5.6.2 Dispositivos Envolvidos

Todos os dispositivos compatíveis com o DLNA atualmente são classificados em quatro categorias diferentes. (SONY,2012).

- **Digital Media Server (DMS):** São dispositivos com a capacidade fornecer aos demais dispositivos as capacidades de receber, armazenar e gravar conteúdo de mídia,

¹Multimedia over Coax Alliance -MoCA é o padrão universal para redes de entretenimento doméstico. usado com cabo coaxial

os próprios DMS podem ter a capacidade de um DMP entre outras funcionalidades de administração de conteúdo multimídia;

- **Digital Media Player (DMP):** São dispositivos com a capacidade de encontrar DMS na rede e reproduzir conteúdos de multimídia;
- **Digital Media Renderer (DMR):** São dispositivos com a capacidade de reproduzir conteúdo multimídia recebido de um DMS proveniente de outro dispositivo da rede como um DMC;
- **Digital Media Controller (DMC):** São dispositivos com a capacidade de controlar o conteúdo multimídia proveniente de um DMS que está sendo reproduzido por um DMP, uma espécie de controle remoto.

5.6.3 Certificação de Dispositivos

Um equipamento só recebe a certificação de compatibilidade DLNA, se passar por uma bateria de testes feitos pelos laboratórios UPnP e Wifi, passando em ambos os laboratórios, como mostra na figura 5.14, o equipamento recebe o certificado, somente assim este equipamento poderá ser comercializado com o logotipo DLNA.

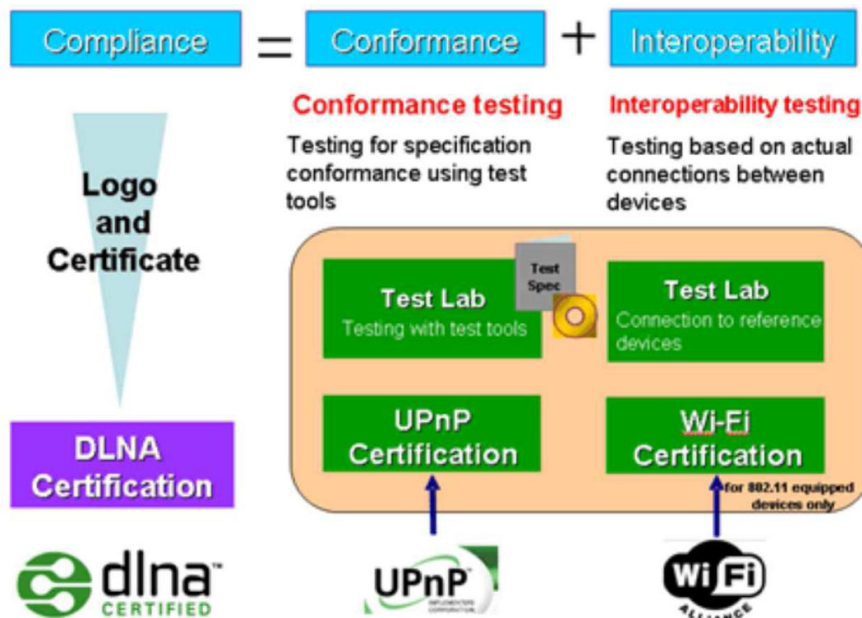


Figura 5.14: Certificação DLNA (SONY,2012)

6 CONCLUSÃO

A tecnologia dos equipamentos computacionais está aumentando o poder de processamento, visto pela evolução dos processadores, podendo assim colocar mais inteligência embarcada, ou seja, dentro de equipamentos simples, não só isso, nas redes também. Juntando essas duas vertentes, equipamentos mais inteligentes e redes residenciais, temos uma casa conectada com equipamentos inteligentes, uma rede com vários serviços disponíveis, favorecendo a descoberta de novas aplicações residenciais.

As pessoas estão cada vez mais conectadas ao mundo, dessa maneira, as tarefas devem ser facilitadas para que as elas possam usar da melhor forma. O Universal Plug and Play é uma iniciativa aberta que objetiva aproveitar padrões, tecnologias e conhecimento existentes, o disponibilizar as pessoas e aos equipamentos novas finalidades com a promessa do mundo conectado sem complicações.

O Universal Plug and Play é baseado em padrões consagrados de protocolos e conceitos sólidos, temos então uma solução aberta e simples, simples o suficiente para funcionar em equipamentos embarcados, como televisão, aparelhos reprodutores de mídia e até celulares, fazendo uma rede entre os dispositivos compartilhando os recursos de uma maneira simples e prática.

A praticidade em usar e em programar tem um grande apelo, o estudo sobre a tecnologia está evoluído, ganhando mais e mais empresas. Para o conceito principal que é o de facilitar as conexões e usar os seus equipamentos conectados e compartilhando recursos, para o ponto de vista do consumidor isso representa muito, no âmbito do ambiente doméstico,

UPnP tem o apoio de grandes corporações e é um protocolo aberto, por isto tem grandes chances de se tornar um padrão bastante usado, o estudo desta tecnologia tem sido base a novas aplicações de comunicação entre equipamentos domésticos.

A tecnologia UPnP resolve vários problemas de conectividade em residências, uma das grandes limitações e referente ao tamanho da rede, se a rede for composta de muitos dispositivos. Quando se conecta muitos dispositivos em um mesmo meio físico de comunicação, aumentará o tráfego de rede, de uma maneira considerável, principalmente

na fase de descobrimento, que normalmente quando se liga os dispositivos, a troca de mensagem é muito intensa. Isso pode causar um atraso temporário na rede.

Na arquitetura UPnP, disponibiliza de forma automática recursos para todos na rede, pode ser levantado o ponto sobre a segurança isso pode se tornar um problema, neste caso poderá fazer uma configuração na rede para aumentar a segurança, mas isso não é automático necessitando de aconselhamento especializado, quando o assunto é segurança sai do conceito de configurações automáticas, existe o risco de arquivos ou pastas compartilhadas ficar disponíveis para outros usuários fora da rede interna, ou seja, poderá está disponível para a internet, para resolver esse tipo de risco é necessário uma configuração de um firewall¹ para isolar a rede doméstica da internet, sendo esta tarefa não tão simples para um usuário comum.

6.1 Avaliação do Trabalho

O objetivo principal, sobre estudar a Arquitetura UPnP, os conceitos relacionados à arquitetura e exemplos de implementação, foram alcançados. Os objetivos específicos, pesquisar e utilizar as ferramentas de desenvolvimento, em específico o Intel UPnP SDK, em criar um dispositivo com a pilha de protocolos padrão UPnP, os objetivos específicos foram alcançados.

A falta de material acadêmico de pesquisa sobre o assunto, arquitetura UPnP, praticamente existe somente um livro sobre detalhamento técnico sobre o assunto, gerou muita dificuldade, a internet ajudou já que existem muitos testes e opiniões sobre essa tecnologia.

A falta de manuais sobre a Intel SDK UPnP, também gerou muita dificuldade, já que foi gasto muito tempo em entender como criar dispositivos e principalmente na criação dos pontos de controle, não foi testado nenhum ponto de controle criado a partir do SDK Intel, até o fim desse trabalho.

¹Firewall é um dispositivo de uma rede de computadores que tem por objetivo aplicar uma política de segurança em um determinado ponto de uma rede.

6.2 Trabalhos Futuros

Como sugestão para trabalhos futuros, a aplicação da tecnologia UPnP, em aparelhos de recepção de televisão digital, aumentando o poder de interatividade com o usuário.

Estudo da aplicação para mídias interativas, exemplo, ao chegar a uma loja, um ponto de controle lê um arquivo de preferências públicas do usuário pelo celular, a propaganda vista por uma tela, vai se modificar de acordo com as suas próprias preferências, de forma automática, se a pessoa gosta de refrigerante, vai ser mostradas marcas de refrigerantes, se a pessoa gosta de cerveja, vai ser mostradas marcas de cerveja e assim sucessivamente.

REFERÊNCIAS BIBLIOGRÁFICAS

BATSYAN, Gena., BOSTANDZHYAN, Sergey., WIMMER, Leonhard., **MediaTomb - UPnP MediaServer** Disponível em: <http://mediatomb.cc/pages/documentation>, acesso em: 20 junho de 2012.

BOLZANI, Caio Augustus Moraes, **Residências Inteligentes**, 1º Edição, São Paulo, ed. Livraria da Física, 2004

BRAY, T. and PAOLI, J. and SPERBERG-McQUEEN and MALER, E., **Extensible Markup Language (XML) 1.0**, Fifth Edition, W3C Recommendation, 2008, Disponível em: <http://www.w3.org/TR/REC-xml>. Acesso em: 16 junho. 2012.

CASTRO, E., **XML para a Word Wide Web**, Ed. Campus, 2001

COMER, Douglas E., **Internetworking with TCP/IP , Volume I , Principles, Protocols and Architecture**, Third Edition, Prentice Hall, 1995.

DLNA, **DLNA - Digital Living Network Alliance** Disponível em: <http://dlna.org/>, acesso em: 19 junho de 2012.

FURGERI, S. **Ensino Didático da Linguagem XML: Aprenda a criar padrões e documentos inteligentes com a XML**. Editora Erica, São Paulo: 2001.

HEITLINGER, Paulo., **O guia Prático da XML**, Coleção: Tecnologias, Centro Atlântico, 2001.

INTEL, **Programming Guide, Intel SDK for UPnP Devices**, Version 1.2.1, 2002.

JERONIMO, M. and WEAST, J. , **UPnP design by example: a software developer's**

guide to universal Plug and Play, Serie: Engineer to Engineer, Ed: Intel Press, 2003.

LIGHT, R., **Iniciando em XML**, São Paulo:Ed. Makron Books, 1999. Livraria da Física, 2004.

RFC1112 Deering, Steve. **RFC 1112 - Host Extensions for IP Multicasting**. Stanford University, 1989.

RFC3986 Berners-Lee, T., Fielding, R., and L. Masinter, **RFC3986 - Uniform Resource Identifier (URI): Generic Syntax**, STD 66, RFC 3986, January 2005.

SONY, **DLNA Home Network**, disponível em http://www.sony.net/SonyInfo/technology/technology/theme/dlna_01.html acesso: 19 de junho de 2012.

TANENBAUM, A.S., **Redes de Computadores** , Editora: CAMPUS, 2003.

ANEXOS

ANEXO A - Código em C# LâmpadaIP criado pela Intel SDK UPnP

```
C:\Users\Alyston\Desktop\Teste\Main.cs 1  
  
// UPnP .NET Framework Device Stack, Core Module  
// Device Builder Build#1.0.4511.28314  
  
using System;  
using OpenSource.UPnP;  
using OpenSource.DeviceBuilder;  
  
namespace OpenSource.DeviceBuilder  
{  
    /// <summary>  
    /// Summary description for Main.  
    /// </summary>  
    class SampleDeviceMain  
    {  
        /// <summary>  
        /// The main entry point for the application.  
        /// </summary>  
        [STAThread]  
        static void Main(string[] args)  
        {  
            // Starting UPnP Device  
            System.Console.WriteLine("UPnP .NET Framework Stack");  
            System.Console.WriteLine("Device Builder Build#1.0.4511.28314");  
            SampleDevice device = new SampleDevice();  
            device.Start();  
            System.Console.WriteLine("Press return to stop device.");  
            System.Console.ReadLine();  
            device.Stop();  
        }  
    }  
}
```

```

C:\Users\Alyston\Desktop\Teste\SampleDevice.cs 1
// UPnP .NET Framework Device Stack, Device Module
// Device Builder Build#1.0.4511.28314

using System;
using OpenSource.UPnP;
using OpenSource.DeviceBuilder;

namespace OpenSource.DeviceBuilder
{
    /// <summary>
    /// Summary description for SampleDevice.
    /// </summary>
    class SampleDevice
    {
        private UPnPDevice device;

        public SampleDevice()
        {
            device = UPnPDevice.CreateRootDevice(1800,1.0,"");

            device.FriendlyName = "LampadaIP";
            device.Manufacturer = "OpenSource";
            device.ManufacturerURL = "http://opentools.homeip.net";
            device.ModelName = "Sample Auto-Generated Device";
            device.ModelDescription = "Sample UPnP Device Using Auto-Generated UPnP Stack";
            device.ModelNumber = "X1";
            device.HasPresentation = false;
            device.DeviceURN = "urn:schemas-upnp-org:device:Sample:1";
            OpenSource.DeviceBuilder.DvImportedService ImportedService = new OpenSource.DeviceBuilder.
DvImportedService();
            ImportedService.External_SwitchPower = new OpenSource.DeviceBuilder.DvImportedService.
Delegate_SwitchPower(ImportedService_SwitchPower);
            device.AddService(ImportedService);

            // Setting the initial value of evented variables
            ImportedService.Evented_Status = false;
        }

        public void Start()
        {
            device.StartDevice();
        }

        public void Stop()
        {
            device.StopDevice();
        }

        public void ImportedService_SwitchPower(System.Boolean St)
        {
            Console.WriteLine("ImportedService_SwitchPower(" + St.ToString() + ")");
        }
    }
}

```


C:\Users\Alyston\Desktop\Teste\DvImportedService.cs

1

```

using OpenSource.UPnP;

namespace OpenSource.DeviceBuilder
{
    /// <summary>
    /// Transparent DeviceSide UPnP Service
    /// </summary>
    public class DvImportedService : IUPnPService
    {
        // Place your declarations above this line

        #region AutoGenerated Code Section [Do NOT Modify, unless you know what you're doing]
        //{{{ Begin Code Block

        private _DvImportedService _S;
        public static string URN = "urn:schemas-upnp-org:service:1";
        public double VERSION
        {
            get
            {
                return(double.Parse(_S.GetUPnPService().Version));
            }
        }

        public delegate void OnStateVariableModifiedHandler(DvImportedService sender);
        public event OnStateVariableModifiedHandler OnStateVariableModified_Status;
        public System.Boolean Evented_Status
        {
            get
            {
                return((System.Boolean)_S.GetStateVariable("Status"));
            }
            set
            {
                _S.SetStateVariable("Status", value);
            }
        }
        public UPnPModeratedStateVariable.IAccumulator Accumulator_Status
        {
            get
            {
                return(((UPnPModeratedStateVariable)_S.GetUPnPService()).GetStateVariableObject("Status")).
                Accumulator;
            }
            set
            {
                ((UPnPModeratedStateVariable)_S.GetUPnPService()).GetStateVariableObject("Status").
                Accumulator = value;
            }
        }
        public double ModerationDuration_Status
        {
            get
            {
                return(((UPnPModeratedStateVariable)_S.GetUPnPService()).GetStateVariableObject("Status")).
                ModerationPeriod;
            }
            set
            {
                ((UPnPModeratedStateVariable)_S.GetUPnPService()).GetStateVariableObject("Status").
                ModerationPeriod = value;
            }
        }
        public delegate void Delegate_SwitchPower(System.Boolean St);

        public Delegate_SwitchPower External_SwitchPower = null;

        public void RemoveStateVariable_Status()
        {
            _S.GetUPnPService().RemoveStateVariable(_S.GetUPnPService().GetStateVariableObject("Status"));
        }
        public void RemoveAction_SwitchPower()

```

SDK para Arquitetura UPnP

Existem várias implementações de SDK para UPnP. São listadas abaixo somente algumas para exemplificar a variedade de ferramentas disponíveis.

Allegro Software

Allegro Software oferece dois diferentes kits de ferramentas para desenvolvimento UPnP, ambos são em ANSI-C e são código aberto. Allegro's Toolkits usa uma camada de abstração de software especial Allegro que permite que ambos os conjuntos de ferramentas possa executar uma grande variedade de plataformas embarcadas, incluindo ATI Núcleo, Logic Express ThreadX, Wind River VxWorks, e todas as plataformas Microsoft Windows. (JERONIMO e WEAST, 2003).

O kit de ferramentas RomUPnP Basic fornece descoberta completa e a capacidade de apresentação em menos de 10 kilobytes de código (não incluindo o servidor web). O kit de ferramentas básico RomUPnP interage com os pontos de controle UPnP 1.0.(JERONIMO e WEAST, 2003).

Lantronix

Novos Kit's de adaptadores fornecem endereçamento e capacidade de descoberta em uma pequena biblioteca em código C. Base adequada para a integração em qualquer um dos Lantronix usado em plataformas embarcadas. O EAK inclui código de exemplo em XML que ilustram como usar.

O código inclui um exemplo em XML que ilustra como usar o Kit de Desenvolvimento com seus dispositivos habilitados para UPnP.(JERONIMO e WEAST, 2003).

Ferramentas de Desenvolvimento SDK Intel

O SDK da Intel para UPnP é uma implementação em .NET Framework, suporta todas as fases de desenvolvimento de dispositivos UPnP. O SDK é disponibilizado

com seu código fonte sobre a licença da Berkley Standart Distribution, atualmente esta disponível na SourceForge, como um projeto de desenvolvimento em código aberto para Linux. A Intel SDK para UPnP Media Renderes está disponível para implementação de Dispositivos UPnP A/V.(JERONIMO e WEAST, 2003).

As ferramentas ajudam os desenvolvedores de software e designers de hardware acelerar o desenvolvimento, testes e implantação de dispositivos compatíveis com UPnP, com ênfase em áudio e distribuição de vídeo. Esse SDK Intel tem as seguintes ferramentas, em código aberto:(INTEL,2002).

- **Device Spy:** Intel Ponto de Controle Universal (UCP). Essa ferramenta disponibiliza a possibilidade de teste com as invocações serviços e eventos.
- **Device Sniffer:** Permite a captura de pacotes de descoberta UPnP. Device Sniffer envia solicitações de pesquisa, as respostas de rastreamento e notificações.
- **Device Validator:** É uma ferramenta que possui um conjunto completo de testes para dispositivos UPnP. São testes de funcionalidades básicas para os dispositivos comuns e mais um conjunto de testes de AV(Áudio e Vídeo) para servidor de conteúdo e processadores de mídia.
- **Device Author:** Cria rapidamente as descrições de dispositivos UPnP sem escrever XML. Essa ferramenta gera automaticamente a descrição de serviço XML válida. Ele também permite importar um serviço de descrição XML de qualquer dispositivo UPnP na disponível rede.
- **Device Relay:** Permite que os dispositivos UPnP conectados em uma rede possam ser disponibilizados em outra. Device Relay espelha os dispositivos UPnP através de uma interface WAN.
- **Network Light:** Exemplo de dispositivo do SDK Intel implementa os serviços SwitchPower e Dimmer. Usa-se Network Light como um dispositivo de referência para invocações e eventos.
- **AV Media Controller:** Ponto de Controle Universal da Intel SDK para dispositivos UPnP AV. AV Media Controller estabelece conexões entre AV UPnP Media Servers e Processadores de Mídia.

- **AV Media Server:** Um servidor de mídia AV configurado para compartilhar arquivos de mídia local. AV Media Server lê os metadados de tags de áudio e formatos de imagem, e torná-lo disponíveis na rede.
- **AV Renderer:** Adiciona um rico conjunto de recursos audiovisuais para controle usa Windows Media Player ActiveX . AV Renderer suporta múltiplas conexões, tipos de mídia, e listas de reprodução.
- **AV Wizard:** Um ponto de controle de implementação leve para dispositivos UPnP AV.