

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Leonardo Maccari Rufino

**INTEGRAÇÃO DO PROTOCOLO SIP À NORMA IEEE 1451 PARA
REDES DE SENSORES SEM FIO**

Florianópolis

2012

Leonardo Maccari Rufino

**INTEGRAÇÃO DO PROTOCOLO SIP À NORMA IEEE 1451 PARA
REDES DE SENSORES SEM FIO**

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Antônio Augusto Medeiros Fröhlich

Florianópolis

2012

Leonardo Maccari Rufino

INTEGRAÇÃO DO PROTOCOLO SIP À NORMA IEEE 1451 PARA REDES DE SENSORES SEM FIO

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Ciência da Computação”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 8 de março de 2012.

Prof. Dr. Ronaldo dos Santos Mello
Coordenador do Curso

Prof. Dr. Antônio Augusto Medeiros Fröhlich
Orientador

Banca Examinadora:

Prof. Dr. Aldri Luiz dos Santos
Presidente

Prof. Dr. João Bosco Mangueira Sobral

Prof. Dr. Roberto Willrich

RESUMO

Redes de sensores sem fio (RSSF) são compostas por dispositivos chamados nós sensores, os quais são capazes de monitorar alguns fenômenos do meio ambiente que os rodeia, tais como informações escalares (temperatura, aceleração) ou multimídia (áudio, vídeo), transformando-os em sinais digitais e comunicando-se com outros nós da rede. A fim de padronizar o acesso e o comportamento das diversas plataformas existentes, a família de padrões IEEE 1451 foi desenvolvida. Esta padronização introduz conceitos interessantes, como a divisão do sistema em duas partes principais, NCAP (*Network Capable Application Processor*) e TIM (*Transducer Interface Module*), e a definição dos TEDS (*Transducer Electronic Data Sheet*). Porém, o padrão não trata eficientemente os requisitos das RSSF atuais, tal como a necessidade dos sensores executarem de forma eficiente e energeticamente consciente, permitindo economizar sua energia, fator crítico em grande parte destes dispositivos. Assim, este trabalho apresenta um novo modo de execução chamado TIM-IM (*TIM Initiated Message*), o qual permite que TIMs reportem seus dados sempre que houver novas leituras sensorizadas, ao invés de aguardar por polling originado pelo NCAP, evitando permanecer com o módulo de comunicação ligado grande parte do tempo. Adicionalmente, o padrão IEEE 1451 limita-se às redes de sensores que captam informações escalares. Assim, a presente dissertação visa, também, a integração de sensores multimídia à norma, apresentando algumas modificações tanto nos TEDS quanto nas mensagens trafegadas entre NCAP e TIM. A fim de permitir o acesso aos sensores através da rede do usuário, foi utilizado o protocolo SIP (*Session Initiation Protocol*). SIP vem sendo bastante utilizado atualmente junto à tecnologia VoIP (*Voice over Internet Protocol*), sendo responsável por estabelecer, modificar e finalizar uma sessão. Devido ao seu tamanho, torna-se inviável seu uso em muitos sistemas embarcados com restrição de recursos. Logo, este trabalho apresenta uma miniaturização do mesmo, alcançada através da eliminação de algumas requisições e campos de cabeçalho (do inglês *header fields*). Por fim, é apresentada a integração do protocolo SIP ao IEEE 1451. Para isto, foi utilizado o estabelecimento de sessões, assim como o esquema de notificação de presença presente no SIP e a extensão relativa à transferência de mensagens instantâneas. Assim, com a união de ambas as normas, permite-se que sensores sejam acessados por usuários remotos utilizando SIP phones, através da Internet, independentemente de sua localização física.

Palavras-chave: Redes de sensores sem fio, SIP, IEEE 1451

ABSTRACT

Wireless sensor networks (WSN) are formed by devices called sensor nodes capable of monitoring some phenomena around them, such as scalar information (temperature, acceleration) or multimedia (audio, video), transforming them into digital signals and communicating with other nodes. In order to standardize the access and behavior of the various platforms available, the IEEE 1451 standards family was developed. This standardization introduces interesting concepts, such as splitting the system into two major parts, NCAP (*Network Capable Application Processor*) and TIM (*Transducer Interface Module*), and the definition of TEDS (*Transducer Electronic Data Sheet*). However, the standard does not address efficiently the requirements of current WSN, such as the need for sensors perform efficiently and energy-conscious, saving its energy, which is critical for most of these devices. This work presents a new execution mode called TIM-IM (*TIM Initiated Message*), which allows TIMs to report its data whenever there are new sensed readings, rather than wait for polling originated by NCAP, avoiding remain with the communication module connected all the time. Additionally, IEEE 1451 is limited to sensor networks that collect scalar information. Thus, this thesis also aims at the integration of multimedia sensors to the standard, presenting some modifications in TEDS and in the messages sent between NCAP and TIM. In order to allow the access to sensors via user's network, it was used the SIP (*Session Initiation Protocol*) protocol. SIP has been widely used today by the VoIP (*Voice over Internet Protocol*) technology and it is responsible to establish, modify and terminate a session. Due to its size, its use is not feasible in many resource-constrained embedded systems. Thus, this work presented a miniaturization of the protocol, achieved through the elimination of some requests and header fields. Finally, it was presented the integration of SIP to IEEE 1451. For this, it was used the session establishment, as well as the presence notification scheme of the SIP protocol and the extension for the transfer of instant messages. Thus, with the union of both standards, sensors can be accessed by remote users using SIP phones through the Internet, regardless of their physical location.

Keywords: Wireless Sensor Network, SIP, IEEE 1451

LISTA DE FIGURAS

Figura 1	Abordagem apresentada pelo presente trabalho.....	22
Figura 2	Os componentes de um nó sensor.	28
Figura 3	EPOSMoteII. Na esquerda, o módulo de sensoriamento. Na direita, o módulo principal.	30
Figura 4	(a) Modelo de transdutor inteligente. (b) Modelo de transdutor inteligente IEEE 1451.	36
Figura 5	Modelo de referência IEEE 1451.	39
Figura 6	Visão alternativa ao modelo de referência IEEE 1451 (IEEE 1451.0, 2007).	40
Figura 7	Modelo de camadas do protocolo SIP.	45
Figura 8	Estabelecimento de uma sessão SIP.	51
Figura 9	Finalização de uma sessão SIP.	53
Figura 10	Inicialização de uma sessão SIP sendo cancelada.	54
Figura 11	Registro de um usuário SIP.	55
Figura 12	Um usuário SIP consultando sobre as capacidades de outro. ...	55
Figura 13	Inscrição e notificação de eventos utilizando SIP.	57
Figura 14	Envio de uma mensagem instantânea utilizando SIP.	58
Figura 15	Arquitetura de camadas do framework orientado a objetos para o padrão IEEE 1451.1 (LEE; SONG, 2005).	64
Figura 16	Acesso HTTP a um NCAP (IEEE 1451.0, 2007).	65
Figura 17	Inicialização e finalização de uma sessão SIP.	68
Figura 18	Inicialização e finalização de uma sessão SIP no Wireshark. ...	71
Figura 19	Visão geral da estratégia tradicional (polling) do padrão IEEE 1451.	72
Figura 20	Visão geral da estratégia TIM-IM para RSSF.	73
Figura 21	Formato da mensagem TIM-IM.	74
Figura 22	Diagrama de sequência para o modo polling.	76
Figura 23	Diagrama de sequência para o modo TIM-IM.	77
Figura 24	Diagrama de sequência para o modo TIM-IM otimizado.	78
Figura 25	Plataforma EPOSMoteII.	79
Figura 26	Formato da mensagem TIM-IM preenchido.	80
Figura 27	Cenário de teste para o modo TIM-IM.	80
Figura 28	Consumo de energia para o sensor de temperatura.	81

Figura 29 Diagrama de sequência para o modo polling utilizando sensor multimídia.	86
Figura 30 Diagrama de sequência para o modo polling otimizado utilizando sensor multimídia.	87
Figura 31 Diagrama de sequência para o modo TIM-IM utilizando sensor multimídia.	88
Figura 32 Diagrama de sequência para o modo TIM-IM otimizado utilizando sensor multimídia.	89
Figura 33 Plataforma EPOSMoteII com microfone.	90
Figura 34 Consumo de energia para o sensor multimídia.	91
Figura 35 Diagrama de sequência para notificação de estado (<i>on-line</i> ou <i>off-line</i>).	93
Figura 36 Diagrama de sequência para sensores não-multimídia utilizando o modo polling.	95
Figura 37 Diagrama de sequência para sensores não-multimídia utilizando o modo TIM-IM.	95
Figura 38 Diagrama de sequência para sensores multimídia utilizando o modo polling.	98
Figura 39 Diagrama de sequência para sensores multimídia utilizando o modo TIM-IM.	99
Figura 40 Diagrama de sequência para cancelamento da chamada em sensores multimídia utilizando o modo TIM-IM.	101
Figura 41 SIP phone Jitsi conectado a um sensor de temperatura.	105
Figura 42 SIP phone Jitsi conectado a um sensor de áudio.	107

LISTA DE TABELAS

Tabela 1	TEDS obrigatórios.	37
Tabela 2	TEDS opcionais.	38
Tabela 3	Campos de cabeçalho SIP.	47
Tabela 4	Classes das respostas SIP.	59
Tabela 5	Códigos das respostas SIP.	59
Tabela 6	Tamanho de implementações SIP (em bytes).	71
Tabela 7	Atributos de transmissão de dados.	75
Tabela 8	Tamanho da implementação do WTIM (em bytes).	82
Tabela 9	Estrutura do bloco de dados do TransducerChannel TEDS ...	83
Tabela 10	Comandos relativos ao estado operacional do transdutor.	85
Tabela 11	Tamanho da implementação do WTIM multimídia (em bytes).	92
Tabela 12	Tempo e número de bytes trafegados para a conexão e desconexão de um sensor TIM-IM.	109

LISTA DE ABREVIATURAS E SIGLAS

RSSF	Redes de Sensores Sem Fio	19
WSN	Wireless Sensor Networks	19
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos	20
OSI	Open Systems Interconnection	21
SIP	Session Initiation Protocol	21
IP	Internet Protocol	21
UDP	User Datagram Protocol	21
TCP	Transmission Control Protocol	21
SCTP	Stream Control Transmission Protocol	21
TLS	Transport Layer Security	21
HTTP	Hypertext Transfer Protocol	21
SOAP	Simple Object Application Protocol	21
CAP	Common Alert Protocol	21
IoT	Internet of Things	22
RFID	Radio-Frequency Identification	22
TIM	Transducer Interface Module	23
NCAP	Network Capable Application Processor	23
TEDS	Transducer Electronic Data Sheet	23
QoS	Quality of Service	27
ADC	Analog-to-Digital Converter	28
USB	Universal Serial Bus	29
WMSN	Wireless Multimedia Sensor Network	31
CMOS	Complementary Metal Oxide Semiconductor	31
TDMA	Time Division Multiple Access	31
VAD	Voice Activity Detector	32
EPOS	Embedded Parallel Operating System	33
C-MAC	Configurable Media Access Control	33
SNMP	Simple Network Management Protocol	35
MIB	Management Information Base	35
API	Application Programming Interface	38
WTIM	Wireless Transducer Interface Module	41
SDP	Session Description Protocol	46

RFC	Request for Comments	46
RTP	Real-Time Transport Protocol	46
RTCP	Real-Time Transport Control Protocol	46
UA	User Agent	46
UAC	User Agent Client	46
UAS	User Agent Server	46
URI	Uniform Resource Identifier	47
PIDF	Presence Information Data Format	56
IM	Instant Messaging	58
MIME	Multipurpose Internet Mail Extensions	58
RVCS	Real-Time Visitor Communication Service	61
SCAG	SIP Context-Aware Gateway	62
DPM	Demand-Driven Parsing Method	62
XML	Extensible Markup Language	62
VoIP	Voice over Internet Protocol	65
STWS	Smart Transducer Web Services	65
SOAP	Simple Object Application Protocol	65
WSDL	Web Service Description Language	65
SAWS	Sensor Alert Web Service	66
SANPS	Sensor Alert Notification Producer Service	66
SANCS	Sensor Alert Notification Consumer Service	66
WS	Web Service	66
TIM-IM	Transducer Interface Module Initiated Message	72
ADPCM	Adaptive Differential Pulse-Code Modulation	90

SUMÁRIO

1 INTRODUÇÃO	19
1.1 OBJETIVO	21
1.1.1 Objetivos Específicos	22
1.2 VISÃO GERAL DO TEXTO	23
2 REDES DE SENSORES SEM FIO	25
2.1 APLICAÇÃO	25
2.2 ARQUITETURA	27
2.3 GERENCIAMENTO DE ENERGIA	29
2.4 SENSORES MULTIMÍDIA	31
2.5 SISTEMAS OPERACIONAIS	32
3 PADRÃO IEEE 1451	35
3.1 TIM E NCAP	35
3.2 TEDS	36
3.3 MEMBROS DA FAMÍLIA DE PADRÕES	37
3.4 TRANSDUCERCHANNEL	41
3.5 MODOS DE OPERAÇÃO	42
3.6 COMUNICAÇÃO SEM FIO ENTRE TIM E NCAP	42
4 PROTOCOLO SIP	45
4.1 ENTIDADES	46
4.2 FORMATO DAS MENSAGENS	47
4.3 REQUISIÇÕES	50
4.3.1 INVITE	50
4.3.2 ACK	51
4.3.3 BYE	52
4.3.4 CANCEL	52
4.3.5 REGISTER	53
4.3.6 OPTIONS	55
4.3.7 SUBSCRIBE e NOTIFY	56
4.3.8 MESSAGE	58
4.4 RESPOSTAS	58
4.5 SIP PARA SISTEMAS EMBARCADOS	61
5 TRABALHOS RELACIONADOS	63
5.1 DISCUSSÃO	66
6 INTEGRAÇÃO DO PROTOCOLO SIP À NORMA IEEE 1451	67
6.1 SIP MINIATURA PARA SISTEMAS EMBARCADOS	67
6.1.1 Implementação	68
6.1.2 Avaliação	70

6.2	MELHORIAS NA COMUNICAÇÃO SEM FIO DO IEEE 1451 . .	72
6.2.1	Implementação	73
6.2.2	Avaliação	78
6.3	SUORTE A REDES DE SENSORES MULTIMÍDIA SEM FIO NO IEEE 1451	82
6.3.1	Implementação	83
6.3.2	Avaliação	90
6.4	INTEGRAÇÃO: SIP X IEEE 1451	92
6.4.1	Notificação do Estado de Conectividade	92
6.4.2	Sensores Não-Multimídia	94
6.4.3	Sensores Multimídia	96
6.4.4	Operações sobre TEDS	100
6.4.5	Demonstrações e Resultados	103
7	CONCLUSÃO	111
7.1	TRABALHOS FUTUROS	113
	Referências Bibliográficas	115

1 INTRODUÇÃO

Com o passar dos tempos, o computador tornou-se uma importante ferramenta na vida da população de qualquer ponto do planeta. Cada dia mais os computadores invadem as casas das pessoas, muitas vezes sem que sejam percebidos. É o caso dos sistemas embarcados (ou também chamados de sistemas embutidos, do inglês *embedded systems*), os quais são, segundo Engblom, “um computador que não se parece com um computador” (ENGBLOM, 2002), ou também, melhor explicado por Machado, “construídos com propósitos específicos e pré-definidos e, em função disto, possuem características que favorecem o uso para este propósito e dificultam o uso para outros fins” (MACHADO, 2008).

Sistemas embarcados estão presentes em dispositivos que tenham algum poder de processamento. Exemplos estão por toda parte, como em brinquedos, eletrodomésticos, aparelhos celulares, automóveis, aviões e mais uma diversidade de produtos (RUFINO, 2009). A elevada relevância destes sistemas é retratada por Tennenhouse, o qual afirma que, no ano 2000, 80% dos processadores produzidos eram utilizados em sistemas embarcados e, entre os restantes, apenas 2% destinavam-se a computadores interativos de propósito geral (TENNENHOUSE, 2000).

Devido à miniaturização e ao barateamento dos componentes eletrônicos, extensas pesquisas têm sido realizadas em um ramo específico de sistemas embarcados. Trata-se das redes de sensores sem fio, ou RSSF (do inglês *Wireless Sensor Networks*, WSN), as quais são compostas por micro-sensores sem fio, de baixo consumo de energia e autônomos (ELSON; ESTRIN, 2000) (LIN et al., 1998). Estes micro-sensores são equipados com um módulo de sensoriamento, analógico ou digital, capaz de sensoriar fenômenos físicos, biológicos ou químicos do meio ambiente circundante (por exemplo, temperatura, luz, som, campo magnético), um processador digital para processamento dos sinais recebidos pelos módulos sensores, um módulo para comunicação (tal como rádio de baixa potência) e, por fim, um módulo de energia para que o sensor possa operar (bateria, célula solar, entre outros) (HEINZELMAN, 2000) (WANNER, 2004) (WANNER, 2006).

Os nós sensores, também conhecidos como *nodes*, são colocados em áreas de interesse, onde cada dispositivo é capaz de coletar dados de seu ambiente, obtendo uma visão do local onde o mesmo se encontra. Normalmente, os dados coletados são então pré-processados no próprio sensor e, em seguida, encaminhados para uma estação base através da rede formada pelo conjunto de nós, onde um usuário final poderá acessar as informações coletadas. Redes de sensores sem fio permitem o monitoramento de uma va-

riedade de ambientes, possivelmente inóspitos, os quais incluem segurança doméstica, diagnóstico de falhas de máquinas, detecção química/biológica, área médica e monitoramento de habitats selvagens (HEINZELMAN, 2000) (MAINWARING et al., 2002). Estas aplicações requerem monitoração confiável, precisa, à prova de falhas e, possivelmente, em tempo real. Enquanto isso, a limitada quantidade de energia, juntamente com as restritas capacidades de processamento dos nós, exige um funcionamento eficiente e energeticamente consciente (WANNER, 2004).

No entanto, devido às inúmeras plataformas disponíveis para a construção destas redes, cada uma com seus atributos de hardware e software, o fator heterogeneidade, juntamente com a mobilidade possivelmente presente entre os nós, tornam as redes de sensores sem fio vulneráveis a falhas. A fim de atenuar estes problemas, foi criada a família de padrões IEEE 1451 (TORRI, 2008). O padrão IEEE 1451 é mantido pelo Comitê Técnico em Tecnologia de Sensores TC-9 da Sociedade de Instrumentação e Medidas no Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) e tem como propósito definir um conjunto de interfaces de comunicação aberta, comum e independente de rede, objetivando conectar transdutores (podendo representar tanto sensores como atuadores) a sistemas baseados em microprocessadores, instrumentos e redes de controle, as quais representam a interface com o usuário final (LEE, 2000) (LEE; SONG, 2007) (SONG; LEE, 2006). Através desta padronização, transdutores de diferentes marcas e modelos podem ser conectados ao sistema de forma automática, através de sua auto-identificação.

O presente trabalho visa integrar sensores conformantes com a norma IEEE 1451 às redes de usuário, utilizando o protocolo SIP (*Session Initiation Protocol*). SIP é responsável por estabelecer, modificar e finalizar uma sessão, a qual representa a troca de dados entre uma associação de participantes, como uma chamada de voz e/ou vídeo (ROSENBERG et al., 2002). Este protocolo está localizado na camada de sessão do modelo OSI (*Open Systems Interconnection*) (COULOURIS; DOLLIMORE; KINDBERG, 2007), apesar de alguns autores citarem o mesmo como estando na camada de aplicação devido ao fato de o protocolo possuir diversos outros propósitos, além do simples estabelecimento de sessões, e, também, porque as pilhas dos protocolos TCP/IP e UDP/IP, as quais são normalmente utilizados junto ao SIP para a comunicação, limitam-se a somente cinco níveis, estando a camada de aplicação logo acima da camada de transporte. SIP utiliza o protocolo IP (*Internet Protocol*), presente na camada de rede, para o estabelecimento de chamadas, podendo utilizar diversos protocolos da camada de transporte, tais como UDP (*User Datagram Protocol*), TCP (*Transmission Control Protocol*), SCTP (*Stream Control Transmission Protocol*) e TLS (*Transport Layer*

Security), sendo UDP o mais comumente encontrado.

A norma IEEE 1451 especifica duas maneiras possíveis, ambas opcionais, para acessar transdutores através de uma rede. A primeira delas utiliza o padrão IEEE 1451.1 (IEEE 1451.1, 2000), o qual define um framework contendo uma série de funcionalidades básicas para a comunicação entre a aplicação de controle presente no NCAP e o usuário remoto. A segunda é conhecida como IEEE 1451.0 HTTP (*Hypertext Transfer Protocol*) e foca principalmente no acesso a dados e TEDS dos transdutores, através do protocolo HTTP 1.1. Outros trabalhos externos a norma definem novas abordagens visando acessar transdutores. Song e Lee utilizam mensagens no formato SOAP/XML (*Simple Object Application Protocol/Extensible Markup Language*) (SONG; LEE, 2007b) e uma continuação desta proposta utiliza o protocolo CAP (*Common Alert Protocol*) (LEE; SONG, 2009).

A seguir serão apresentados os objetivos deste trabalho, assim como uma visão geral da presente dissertação.

1.1 OBJETIVO

Este trabalho apresenta uma nova abordagem a fim de acessar transdutores através da rede do usuário. Isto será feito através da integração de sensores sem fio, conformantes com a norma IEEE 1451, a redes SIP, possibilitando a comunicação entre usuários e sensores através da Internet, utilizando um protocolo bastante difundido atualmente, SIP. A utilização deste protocolo traz benefícios em frente às demais abordagens existentes, principalmente quando utilizado sensores multimídia, pois estas não tratam este tipo de dispositivo e SIP representa um protocolo próprio para o estabelecimento de sessões para comunicação interativa entre usuários.

Assim, através de um dispositivo executando uma aplicação SIP, tais como computadores, celulares, PDAs (*Personal Digital Assistant*), entre outros, pode-se ter acesso instantâneo às informações capturadas pelos sensores, como por exemplo, imagens de uma câmera de segurança ou áudio de um microfone. Já a presença da norma IEEE 1451 possibilita que diferentes sensores sejam conectados aos sistemas de forma automática, através de sua auto-identificação e do conjunto de interfaces de comunicação padronizado. A figura 1 apresenta a solução introduzida pela presente dissertação, onde o NCAP exerce o papel de gateway, adaptando mensagens SIP ao formato IEEE 1451 e vice-versa.

A união da abordagem apresentada neste trabalho com o chamado Internet das Coisas (do inglês *Internet of Things*, IoT) parece ser bastante promissora. O conceito de IoT baseia-se, principalmente, em sensores sem

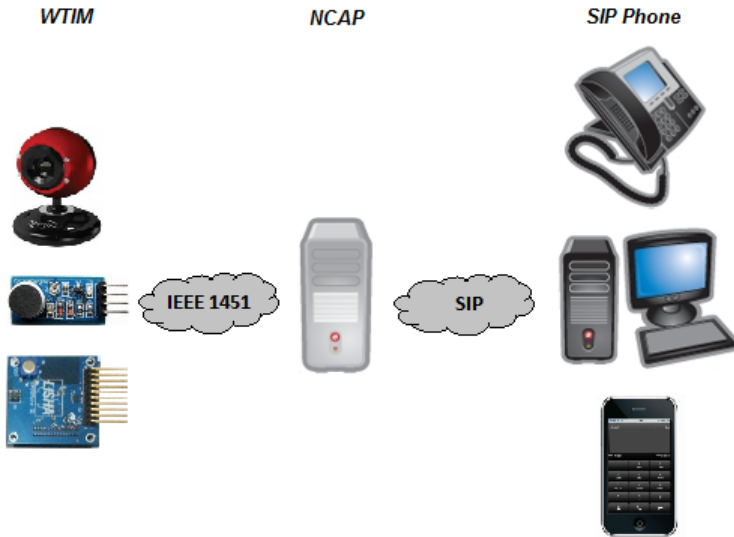


Figura 1: Abordagem apresentada pelo presente trabalho.

fio e na nanotecnologia, e sua ideia principal é permitir a conexão e interação com qualquer objeto, independentemente de seu tamanho físico, utilizando etiquetas RFID (*Radio-Frequency Identification*), as quais usam frequências de rádio para identificação. Com o avanço desta tecnologia, pessoas poderão interagir através de seu ambiente virtual, utilizando o protocolo SIP, a uma diversidade de objetos do mundo real.

1.1.1 Objetivos Específicos

- Apresentar melhorias na comunicação de sensores sem fio conformantes com o padrão IEEE 1451.
- Introduzir o conceito de sensores multimídia sem fio à norma IEEE 1451.
- Apresentar uma versão SIP miniatura, visando sua utilização em sistemas embarcados com restrição de recursos.
- Modelar e implementar a integração entre o padrão IEEE 1451 e o protocolo SIP, focados na área de redes de sensores sem fio.

1.2 VISÃO GERAL DO TEXTO

O capítulo 2 apresenta conceitos relacionados às redes de sensores sem fio, tais como aplicação, arquitetura, gerenciamento de energia, sensores multimídia, assim como os requisitos necessários aos sistemas operacionais os quais operam nos sensores.

Os objetivos da família de padrões IEEE 1451 são apresentados no capítulo 3, descrevendo cada um de seus membros e introduzindo os conceitos TIM (*Transducer Interface Module*), NCAP (*Network Capable Application Processor*), TEDS (*Transducer Electronic Data Sheet*) e *Transducer-Channel*. Adicionalmente, são apresentados os modos de operação existentes para os transdutores, concluindo com a comunicação sem fio entre TIM e NCAP.

O capítulo 4 fornece uma base sobre o protocolo de inicialização de sessão SIP. Este capítulo apresenta as entidades SIP definidas em sua norma, o formato das mensagens, as requisições e suas possíveis respostas. Também, é comentado o uso do protocolo voltado ao setor dos sistemas embarcados.

O capítulo 5 apresenta trabalhos relacionados visando acessar transdutores através da rede do usuário. A integração da norma IEEE 1451 ao protocolo SIP para redes de sensores sem fio é apresentada no capítulo 6. O mesmo discute três passos intermediários essenciais para a execução do objetivo final deste trabalho. O primeiro deles implementa uma versão SIP miniatura visando utilizá-la em sistemas embarcados com restrição de recursos. O segundo passo apresenta melhorias na comunicação sem fio para sensores IEEE 1451. E, por fim, é apresentado o suporte a sensores multimídia ao padrão IEEE 1451. A seguir, é apresentada a solução de integração entre IEEE 1451 e SIP adotada, tanto para sensores escalares quanto para multimídia, finalizando com os resultados da implementação.

Finalmente, o capítulo 7 conclui este trabalho, apresentando, também, algumas sugestões para a continuidade do mesmo.

2 REDES DE SENSORES SEM FIO

Os avanços recentes nas comunicações sem fio e referentes à eletrônica permitiram o desenvolvimento de nós sensores de baixo custo, baixa potência e multifuncionais, os quais são pequenos em tamanho e comunicam-se sem restrições em curtas distâncias. Estes minúsculos nós sensores, capazes de sensoriar o ambiente, processar os dados e comunicar-se entre si, alavancam a ideia de redes de sensores sem fio (AKYILDIZ et al., 2002).

2.1 APLICAÇÃO

Redes de sensores podem ser utilizadas em diversas áreas de aplicação. As pesquisas iniciais neste setor surgiram, principalmente, motivadas por aplicações militares e, mais recentemente, outros domínios de aplicação para redes de sensores sem fio têm sido considerados, tais como monitoração de ambientes e espécies, na área da saúde, agricultura, entre outros.

Na área militar, por exemplo, a rápida implantação, auto-organização e as características de tolerância a falhas de redes de sensores tornam-nas uma técnica muito promissora para sensoriamento de comando militar, comunicação, vigilância, reconhecimento e sistemas de identificação. Na saúde, os sensores também podem ser implantados a fim de monitorar pacientes e ajudar os portadores de deficiência. Algumas outras aplicações comerciais incluem a gestão de inventário, monitoramento da qualidade de produtos e supervisão de áreas de desastres (AKYILDIZ et al., 2002).

Sabendo que as redes de sensores sem fio estão situadas em uma ampla variedade de aplicações e sistemas, com requisitos e características muito diferentes entre si, Römer propõe uma caracterização quanto às várias dimensões existentes no espaço de projeto de redes de sensores (RÖMER; MATTERN, 2004). Esta classificação é dada por doze dimensões principais, as quais serão discutidas abaixo:

- **Implantação:** Nós podem ser aleatoriamente dispersos em uma zona de cobertura ou cada dispositivo pode ser estrategicamente colocado em um local específico. Em algumas situações, tais como ambientes adversos, só há uma oportunidade para implantar os sensores, enquanto que em outros casos, os nós podem ser atualizados e substituídos assim que necessário, por exemplo, quando os mesmos ficarem sem energia.
- **Mobilidade:** Sensores podem variar entre totalmente imóveis, movidos por eventos naturais, ou até mesmo estando em constante movi-

mento.

- **Custo, Tamanho, Recursos e Energia:** Estes fatores são dependentes dos requisitos da aplicação. Os nós de uma rede podem variar em preço, podendo custar várias centenas de dólares ou apenas alguns centavos cada. Alguns são tão grandes como uma caixa de sapato, enquanto pesquisas atuais tentam desenvolver nós tão pequenos quanto possível. Por fim, alguns sensores são limitados por pequenas baterias, enquanto que outros extraem energia a partir de alguma outra fonte, como é o caso dos painéis solares.
- **Heterogeneidade:** O padrão original para redes de sensores era composto de dispositivos homogêneos, sem características de identificação única. Isso tem mudado e, atualmente, há redes heterogêneas com nós especiais possuindo propósitos específicos de serviço. Por exemplo, alguns nós podem possuir um sensor GPS e outros podem incluir as interfaces para redes conectadas à Internet.
- **Modalidade de Comunicação:** Enquanto a frequência por ondas de rádio é a modalidade de comunicação mais comum, a informação pode também ser transmitida através de outras formas, tais como laser, som e luz difusa.
- **Infraestrutura:** As redes de sensores sem fio podem suportar uma variedade de infraestruturas de rede. Por exemplo, em uma rede baseada em infraestrutura, os nós podem somente se comunicar com uma estação base. Já em uma rede ad-hoc, não há uma estação base ou uma infraestrutura de comunicação, assim, cada nó pode se comunicar com qualquer outro dispositivo e os mesmos podem agir como roteadores, encaminhando mensagens em nome de outros sensores.
- **Topologia de Rede:** Uma propriedade importante de uma rede de sensores é o seu diâmetro, ou seja, o número máximo de saltos (*hops*) entre quaisquer dois nós da rede. Na sua forma mais simples, uma rede de sensores forma uma rede de salto único (*single-hop*), onde cada nó sensor é capaz de comunicar-se diretamente com todos os outros nós. Já uma rede de múltiplos saltos (*multihop*) pode formar um grafo arbitrário, mas, frequentemente, uma rede com uma estrutura mais simples é construída, tal como uma árvore ou um conjunto de redes em estrela conectadas.
- **Cobertura:** Requisitos de cobertura dependem da aplicação. Alguns aplicativos podem alcançar seus objetivos através de uma rede com sensores implantados esparsamente, onde apenas partes da área de inte-

resse são cobertos pelos nós sensores. Outras aplicações exigem uma rede densamente povoada, na qual a área de interesse é completamente coberta pelos sensores, podendo haver nós redundantes disponíveis, fazendo com que vários dispositivos cubram o mesmo local físico.

- **Conectividade:** As faixas de comunicação e localização física dos nós sensores individuais definem a conectividade de uma rede. Caso sempre haja uma conexão de rede entre quaisquer dois nós, a rede é dita ser conectada. A conectividade é intermitente se a rede pode ser ocasionalmente particionada. E quando os nós estão isolados a maior parte do tempo e entram no alcance de comunicação de outros sensores apenas ocasionalmente, a comunicação é chamada de esporádica.
- **Tamanho da Rede:** Requisitos de cobertura e topologia da rede determinam o tamanho da mesma. O tamanho da rede pode variar entre milhares de nós a apenas alguns poucos.
- **Tempo de Vida:** Os requisitos da aplicação estabelecem o tempo de vida necessário de uma rede de sensores sem fio. Alguns aplicativos precisam de uma rede que estará disponível por muitos anos após a implantação, outras são necessárias apenas por algumas horas. A energia disponível e a redundância entre os nós influenciam o tempo de vida da rede.
- **Requisitos de QoS:** Dependendo da aplicação, uma rede de sensores deve suportar determinados aspectos de qualidade de serviço (do inglês *Quality of Service*, QoS), tais como restrições de tempo real, robustez, resistência a adulteração (*tamper-resistance*) e resistência a espionagem (*eavesdropping resistance*).

2.2 ARQUITETURA

A principal tarefa de um nó sensor é detectar eventos, executar processamento simples nos dados localmente e, então, transmitir estas informações a outros nós ou a uma estação base (AKYILDIZ et al., 2002). Cada nó individual possui capacidade limitada, mas a comunicação e processamento cooperativo na rede permitem a obtenção de dados mais precisos. Sendo assim, um nó sensor é composto por quatro componentes básicos, como mostrado na figura 2: unidade de sensoriamento, unidade de processamento, unidade transceptora e unidade de energia.

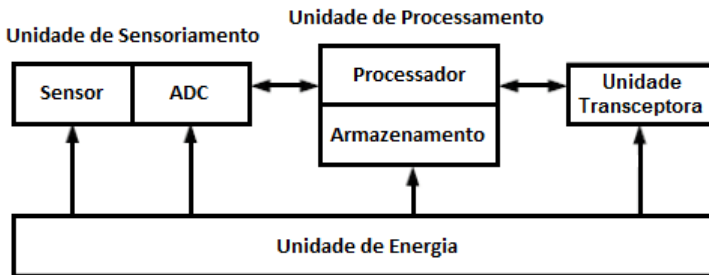


Figura 2: Os componentes de um nó sensor.

As unidades de sensoriamento são geralmente compostas por duas subunidades: sensores e conversores analógico-digital (do inglês *Analog-to-Digital Converter*, ADC). Os sinais analógicos capturados pelos sensores, baseados no fenômeno observado, são convertidos em sinais digitais através do ADC e, em seguida, encaminhados para a unidade de processamento. A unidade de processamento, frequentemente associada a uma unidade de armazenamento pequena, gerencia os processos que compõem o nó sensor, colaborando com os outros nós para a realização das tarefas de sensoriamento atribuídas aos mesmos. Uma unidade transceptora conecta o nó à rede, sendo responsável pela comunicação com os outros nós sensores. Por fim, unidades de energia são responsáveis por fornecer corrente elétrica para todo o sistema e podem ser apoiadas por fontes renováveis, tais como células solares.

Com base nas pesquisas e aplicações atuais, Wanner cita que a arquitetura básica de hardware de um nó sensor deve (WANNER, 2006):

- **Ter dimensões físicas reduzidas:** Para poderem ser instalados de maneira não intrusiva, os nós sensores devem ter dimensões reduzidas. Dado o constante avanço das técnicas de miniaturização de hardware, o tamanho dos componentes eletrônicos utilizados nos nós tende a diminuir constantemente. Entretanto, a miniaturização dos nós sensores pode estar limitada ao tamanho da fonte de energia, seja na forma de baterias ou dispositivos para a captura da energia ambiente.
- **Ser capaz de operar por um longo período de tempo com quantidade limitada de energia:** A necessidade de operação autônoma de um nó sensor e a capacidade limitada de energia disponível ao mesmo fazem com que o baixo consumo de energia seja um fator determinante no projeto de hardware. Sendo assim, o projeto de um nó sensor priorizar componentes de baixa potência e com suporte a gerência do consumo de energia, tais como microcontroladores e transceptores de

baixa potência, em detrimento de componentes direcionados à alta capacidade de processamento, desempenho ou potência.

- **Ter um projeto modular, permitindo a conexão com sensores específicos para diferentes aplicações:** Os serviços de uma rede de sensores tendem a ser específicos e visam utilizar somente o hardware necessário aos requisitos de cada aplicação (HILL et al., 2000). Desta forma, é importante que o projeto seja modular e permita a remoção e inclusão de sensores conforme as necessidades da aplicação.
- **Permitir a mais ampla configuração possível do canal de transmissão de dados:** O transceptor de dados sem fio é, em geral, o componente com maior consumo de energia em um nó sensor (LANGENDOEN; HALKES, 2005). Assim, é importante que este transceptor passe a maior parte do tempo desligado. Por outro lado, diferentes aplicações terão padrões de comunicação específicos e poderão se beneficiar de diferentes técnicas de modulação de dados e controle de acesso ao meio, permitindo o controle do consumo de energia sem comprometer a comunicação de dados. Consequentemente, o transceptor deve permitir a maior configuração possível do canal de dados.

EPOSMoteII (LISHA, 2010) representa um exemplo de um nó sensor, o qual implementa todas as funções básicas apresentadas anteriormente. Esta plataforma foca em modularização e é composta por módulos intercambiáveis para cada função (processamento, comunicação, sensoramento e fornecimento de energia). A figura 3 apresenta o kit de desenvolvimento, o qual é ligeiramente maior do que uma moeda de R\$1. O módulo principal é responsável pelo processamento, armazenamento e comunicação. EPOSMoteII possui um processador ARM de 32 bits, 128kB de flash, 96kB de RAM e um módulo de rádio compatível com IEEE 802.15.4. O módulo de sensoramento padrão contém um sensor de temperatura e um acelerômetro, leds, switches e uma porta micro USB (*Universal Serial Bus*), podendo ser utilizada como fonte de energia.

2.3 GERENCIAMENTO DE ENERGIA

Nós sensores devem distribuir o consumo de energia entre os modos de dormência, sensoramento, cálculo local e comunicação. Porém, como os diferentes nós da rede possuem diferentes funções, os mesmos podem também ter requisitos de energia muito diferentes. Por exemplo, nós localizados próximo ao gateway da rede podem precisar encaminhar todas as mensagens recebidas a partir de nós sensores distantes, ao passo que um nó em

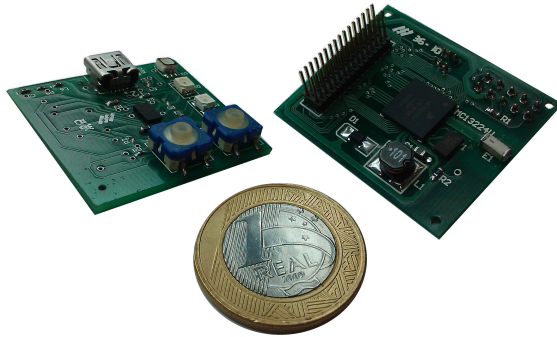


Figura 3: EPOSMoteII. Na esquerda, o módulo de sensoriamento. Na direita, o módulo principal.

um extremo da rede pode necessitar apenas relatar as suas próprias leituras (MAINWARING et al., 2002). Ou seja, em muitas redes, há um conjunto de nós com energia limitada e quando estes dispositivos esgotam suas fontes de alimentação, a rede pode se tornar desconectada e inoperável. Consequentemente, é preciso orçar a energia, principalmente no que diz respeito ao gargalo da rede.

Como a principal tarefa de um nó sensor é detectar eventos, executar processamentos nestes dados e transmiti-los, o consumo de energia pode ser, então, dividido em três domínios: sensoriamento, processamento de dados e comunicação (AKYILDIZ et al., 2002). O tempo de vida base de um nó é determinado pelo consumo da corrente no estado de dormência. A minimização de energia em modo de hibernação envolve desativar os sensores, o rádio e colocar o processador em modo de sono profundo (MAINWARING et al., 2002).

Sabendo que a maioria dos sensores necessita de uma fonte de energia para energizar o elemento sensor e os componentes eletrônicos associados, tais como energia elétrica ou bateria, Gilsinn e Lee comentam que sensores sem fio com restrito estoque de energia seriam úteis em locais onde a manutenção é realizada regularmente, podendo, assim, substituir tais baterias. Também, outra maneira de energizar sensores seria através da extração de energia do seu meio circundante. Desta forma, um dispositivo de conversão de energia seria destinado a capturar energia do calor, luz, som, vibração ou qualquer outra fonte disponível no ambiente (GILSINN; LEE, 2001).

2.4 SENSORES MULTIMÍDIA

Enquanto que as redes de sensores sem fio tradicionais são formadas por nós sensores capazes de monitorar algum fenômeno físico escalar em seu meio circundante (tais como temperatura, pressão, umidade), as redes de sensores multimídia sem fio (do inglês *Wireless Multimedia Sensor Network*, WMSN) possuem sensores mais elaborados. Esta nova tecnologia surgiu durante os últimos anos, alavancada pela produção de câmeras e microfones baseados em CMOS (*Complementary Metal Oxide Semiconductor*) baratos. Estes sensores multimídia podem adquirir um rico conteúdo do ambiente, por exemplo, vídeo e áudio. WMSN pode ser definida como uma rede possuindo nós sensores interconectados através de um enlace sem fio, os quais são equipados com dispositivos multimídia, como câmeras, e que são capazes de capturar fluxos de vídeo e áudio, imagens e dados de sensores escalares (ALNUAIMI; SALLABI; SHUAIB, 2011).

Um exemplo de rede de sensores sem fio capaz de trafegar fluxos de voz em tempo real é descrito por Mangharam. Este artigo apresenta o chamado FireFly, uma plataforma para redes de sensores multimídia sincronizada com relação ao tempo e que utiliza o sistema operacional de tempo real para sensores Nano-RK (ESWARAN; ROWE; RAJKUMAR, 2005). A fim de alcançar alta vazão fim-a-fim, latência limitada e duração de vida previsível, foi empregado um sincronizador de tempo baseado em hardware. O autor também propõe usar RT-Link (ROWE; MANGHARAM; RAJKUMAR, 2006), um protocolo da camada de enlace baseado em TDMA (*Time Division Multiple Access*), para a troca de mensagens em intervalos de tempo bem definidos. Por último, uma implantação experimental de 42 nós em uma mina de carvão é descrita, apresentando medidas de desempenho relacionadas à vazão fim-a-fim, variação de atraso, perda de pacotes e qualidade da voz (MANGHARAM et al., 2006).

O eficiente uso da largura de banda em redes de sensores sem fio é de interesse devido às restrições impostas pelo seu tamanho e potência. Em aplicações onde um sinal de áudio deve ser transmitido a partir de sensores a uma estação base através de um meio sem fio, é interessante parar a transmissão quando não há a presença de voz. Como um pré-processador para tal sistema, um detector de atividade de voz (do inglês *Voice Activity Detector*, VAD) é requerido para decidir se o áudio atual deve ser transmitido ou não. Berisha propõe um VAD de baixa complexidade e um classificador de sexo (masculino ou feminino) simples para um sistema de sensoriamento acústico distribuído. Uma decisão local é tomada pelo sensor, baseado nas medições realizadas e, então, os dados podem ser transmitidos para a estação base. Na estação base, um algoritmo de fusão de dados é implementado, o qual com-

bina as decisões individuais dos sensores localizados na rede para formar uma decisão final (BERISHA; KWON; SPANIAS, 2006).

A compressão de dados também é um aspecto importante em WMSN devido a sua limitada largura de banda disponível para comunicação entre sensores sem fio. Gosztolya propõe usar a codificação de Huffman, um conhecido método de compressão sem perdas e utilizado para qualquer tipo de dados, pré-calculando os códigos de Huffman para cada possível amostra. Desta forma, o codificador deve, somente, ler os códigos apropriados em uma tabela pré-calculada para cada valor de amostra. Utilizando Huffman combinado com outros métodos de compressão de áudio, foi obtida uma significativa redução na largura de banda requerida, tal como 30% usando o codec μ -law, com praticamente nenhuma perda de informação, e 50%, com uma pequena perda de informação, usando o método de truncagem (GOSZTOLYA; PACZOLAY; ANDTH, 2010).

2.5 SISTEMAS OPERACIONAIS

Uma decisão importante ao projetar uma rede de sensores é a escolha do sistema operacional que executará em cada nó sensor. Estes sistemas devem ser cuidadosamente desenvolvidos, pois os mesmos, além de exercerem o controle do hardware, devem abstraí-los, fornecendo uma camada de mais alto nível ao desenvolvedor da aplicação, ao mesmo tempo em que realizam a gerência dos recursos limitados do dispositivo.

Wanner comenta que os requisitos dos sistemas operacionais para redes de sensores sem fio são bastante amplos. Algumas características que os mesmos devem possuir para operar de maneira satisfatória são: incluir a funcionalidade básica de um sistema operacional, executar serviços de gerência relacionados ao consumo de energia, prover mecanismos para reprogramação, fornecer uma abstração de hardwares heterogêneos, possuir uma pilha de comunicação configurável e, por fim, operar com recursos limitados. Sendo assim, torna-se difícil o uso de sistemas operacionais tradicionais em dispositivos sensores (WANNER, 2006).

Desenvolvido originalmente na Universidade da Califórnia em Berkeley, TinyOS é, atualmente, o sistema operacional mais utilizado em redes de sensores. TinyOS representa um sistema operacional de código aberto e escrito na linguagem de programação nesC. O sistema em si consome poucos recursos e é capaz de executar em plataformas de 8 bits com menos de 1kB de RAM. Entretanto, o modelo de concorrência simplificado, a falta de gerência dinâmica de recursos e o modelo de abstração de hardware excessivamente dependente de plataforma fazem com que as aplicações muitas vezes tenham

que completar a funcionalidade do sistema operacional. Isto é evidentemente indesejável, já que compromete a portabilidade das aplicações e a reusabilidade de código, além de agregar tamanho e consumo de recursos ao sistema (WANNER, 2006).

Outro conhecido sistema operacional próprio para sistemas embarcados, inclusive atuando em nós sensores, é o EPOS (*Embedded Parallel Operating System*) (FRÖHLICH, 2001). EPOS é um framework baseado em componentes para a geração de ambientes de suporte de execução dedicados, permitindo que programadores desenvolvam aplicativos independentes de plataforma. O projeto EPOS visa automatizar o desenvolvimento de sistemas embarcados para que desenvolvedores possam se concentrar no que realmente importa: suas aplicações (MARCONDES et al., 2006). EPOS tem um pequeno footprint de memória, uma pilha de comunicação completa e suporte para redes de sensores.

O ambiente de suporte de execução EPOS para aplicações de redes de sensores inclui, além das funções básicas de um sistema operacional executando com recursos limitados, mecanismos de gestão de energia eficiente, reprogramação em campo, abstração uniforme de dispositivos sensores heterogêneos e serviço de comunicação configurável através do protocolo de controle de acesso ao meio C-MAC (*Configurable Media Access Control*), cumprindo assim os requisitos específicos encontrados em aplicações para redes de sensores sem fio (FRÖHLICH; WANNER, 2008).

A utilização do EPOS no projeto do sistema permite que programadores desenvolvam aplicações independentes de plataforma, pois ferramentas de análise permitem a geração de um sistema de suporte de execução que agregue todos os recursos que a aplicação específica necessita, e nada mais. Por definição, uma instância do sistema utiliza somente os recursos necessários ao suporte da aplicação. Ao mesmo tempo, o repositório de componentes do sistema disponibiliza um grande conjunto de serviços presentes em sistemas operacionais tradicionais, através de interfaces independentes de plataforma. EPOS suporta diversas plataformas computacionais heterogêneas, como IA32, PowerPC, H8, Sparc, MIPS e AVR (WANNER, 2006).

3 PADRÃO IEEE 1451

O Comitê Técnico em Tecnologia de Sensores TC-9 da Sociedade de Instrumentação e Medidas no Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) está trabalhando no estabelecimento de um conjunto de padrões para interface de transdutores (sensores e atuadores) inteligentes chamado IEEE 1451. O foco desta família de padrões é definir um conjunto de interfaces de comunicação comum, com o objetivo de conectar transdutores a sistemas baseados em microprocessadores, instrumentos e redes de campo, independentemente do tipo de rede utilizada (LEE,).

As especificações desenvolvidas não colocam restrições com relação ao uso de condicionadores de sinais e sistemas de processamento, conversores analógico-digital e digital-analógico, microprocessadores, protocolos de rede e meios de comunicação. Desta forma, a meta do padrão IEEE 1451 é reduzir o esforço da indústria em desenvolver e migrar para transdutores inteligentes de rede.

Um transdutor inteligente representa a integração de um elemento sensor ou atuador analógico ou digital, uma unidade de processamento, uma interface de comunicação e o software associado para condicionamento de sinal, calibração, diagnóstico e comunicação (SONG; LEE, 2008). Baseado nisso, um modelo de transdutor inteligente é apresentado na figura 4 (a). A figura 4 (b) mostra a arquitetura do transdutor inteligente do padrão IEEE 1451, a qual é bastante similar à anterior. As diferenças básicas estão na divisão do sistema em dois componentes principais, chamados TIM e NCAP, e a introdução dos TEDS. Relacionando a norma IEEE 1451 ao protocolo de gerência de redes SNMP (*Simple Network Management Protocol*), os TEDS exerceriam um papel similar ao das MIBs (*Management Information Base*).

3.1 TIM E NCAP

A fim de atingir intercambialidade transdutores-rede e interoperabilidade transdutor-redes, a família de padrões IEEE 1451 divide o sistema em duas categorias de dispositivos, NCAP (*Network Capable Application Processor*) e TIM (*Transducer Interface Module*).

- **TIM:** É o módulo que contém a interface com o NCAP, o condicionador de sinal, conversor analógico-digital e/ou digital-analógico e, em muitos casos, o transdutor. Um TIM pode variar em complexidade de um único sensor ou atuador a unidades contendo muitos transdutores (sensores e atuadores).

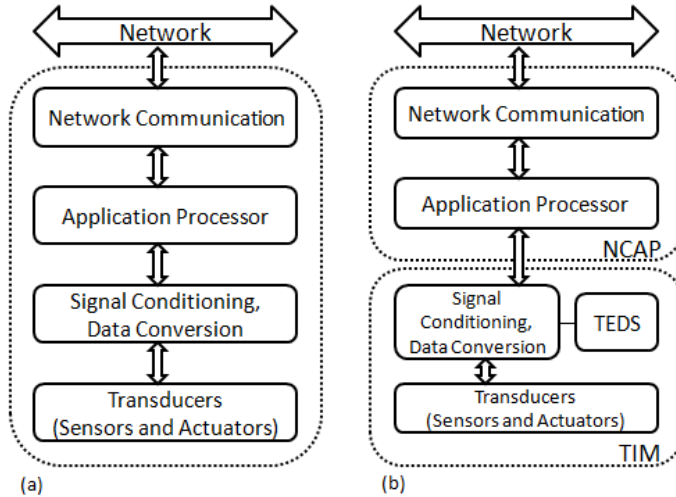


Figura 4: (a) Modelo de transdutor inteligente. (b) Modelo de transdutor inteligente IEEE 1451.

- **NCAP:** É o hardware e software que fornece a função de gateway entre o TIM e a rede do usuário.

Alguns membros da família de normas IEEE 1451 estabelecem a interface de comunicação entre o NCAP e os TIMs conectados a ele (IEEE 1451.0, 2007).

3.2 TEDS

Uma característica chave de um transdutor inteligente IEEE 1451 é a especificação do TEDS (*Transducer Electronic Data Sheet*), o qual contém informações relacionadas ao transdutor, tais como nome do fabricante, tipo de sensor/atuador, número de série, faixa de medição, precisão e dados de calibração, entre outras. Os TEDS fornecem muitos benefícios, por exemplo (SONG; LEE, 2008):

- **Permitem a auto-identificação de sensores ou atuadores:** Um sensor ou atuador equipado com TEDS, seguindo o padrão IEEE 1451, pode se auto-identificar e auto-descrever em frente ao NCAP através do envio das informações contidas nos TEDS.
- **Provêm auto-documentação de longo prazo:** Os TEDS presentes

nos TIMs podem ser atualizados e armazenar informações, tais como a localização do transdutor, data da recalibração, registros de reparos e outros dados relacionados à manutenção.

- **Reduzem erro humano:** A transferência automática dos dados contidos nos TEDS ao NCAP elimina a entrada de parâmetros manuais no sensor, os quais poderiam induzir a erros.
- **Facilitam a instalação em campo, atualização e manutenção dos transdutores:** Isso ajuda a reduzir os custos totais do ciclo de vida de sistemas com transdutores, pois qualquer pessoa pode executar essas tarefas devido ao simples *plug and play* dos sensores.
- **Fornecem capacidade *plug and play*:** TIMs e NCAPs construídos com base no padrão IEEE 1451 estão aptos a serem conectados a um meio de comunicação físico padronizado e são capazes de operar sem qualquer alteração no software do sistema. Não há necessidade de drivers diferentes, perfis ou outras alterações de software a fim de proporcionar as operações básicas dos transdutores.

Há quatro tipos de TEDS mandatórios, ou seja, a norma especifica que os mesmos deverão ser descritos, sendo que os restantes são opcionais. A tabela 1 apresenta os TEDS obrigatórios, enquanto que a tabela 2 informa os TEDS opcionais existentes. Os TEDS podem ser armazenados em algum tipo de ROM programável e eletricamente apagável, caso o conteúdo nunca mude, ou então, as partes mutáveis dos TEDS podem ser colocadas na memória RAM do TIM (SONG; LEE, 2008).

Tabela 1: TEDS obrigatórios.

TEDS Obrigatórios
Meta TEDS
TransducerChannel TEDS
User's Transducer Name TEDS
PHY TEDS

3.3 MEMBROS DA FAMÍLIA DE PADRÕES

Atualmente, a família IEEE 1451 consiste de seis normas ativas e uma em desenvolvimento. O modelo de referência apresentado na figura 5 mostra

Tabela 2: TEDS opcionais.

TEDS Opcionais	
Meta-Identification TEDS	Transfer Function TEDS
TransducerChannel Identification TEDS	Commands TEDS
Calibration TEDS	Location and Title TEDS
Calibration Identification TEDS	Geographic Location TEDS
End Users' Application-Specific TEDS	Units Extention TEDS
Frequency Response TEDS	Manufacturer-Defined TEDS

os relacionamentos entre os vários membros da família e a figura 6 apresenta uma visão alternativa à mesma. Cada membro será descrito sucintamente a seguir.

- **IEEE 1451.0:** Introduz o conceito de TIM e NCAP, conectados através de um meio de comunicação especificado por outro membro da família. Também, descreve um conjunto de funcionalidades comuns a toda norma, as quais são independentes do meio físico de comunicação, e incluem especificações dos formatos de alguns TEDS, protocolos de comunicação comuns e funções básicas necessárias para controlar e gerenciar os transdutores. Além disso, IEEE 1451.0 define um conjunto de APIs (*Application Programming Interface*) independentes de implementação (IEEE 1451.0, 2007).
- **IEEE 1451.1:** Define uma interface para conexão de NCAPs a redes de controle, através do desenvolvimento de um modelo de objeto comum e uma especificação de interface para sensores e atuadores inteligentes. A norma IEEE 1451.1 é aplicável a aplicações de controle e medição distribuída e foca, principalmente, nas comunicações entre os próprios NCAPs, assim como entre NCAPs e os outros nós do sistema (IEEE 1451.1, 2000).
- **IEEE 1451.2:** Estabelece uma interface digital para conexão entre transdutores e microprocessadores. O mesmo descreve alguns TEDS e seus formatos de dados, assim como define uma interface elétrica, funções lógicas de leitura e escrita para acessar os TEDS e uma ampla variedade de transdutores. Esta norma não especifica qual sistema de condicionamento e conversão de sinal deve ser utilizado, nem mesmo como os dados contidos nos TEDS devem ser usados pelas aplicações. Ou seja, o padrão IEEE 1451.2 define uma interface transdutor-NCAP e TEDS para configurações ponto-a-ponto (IEEE 1451.2, 1998).

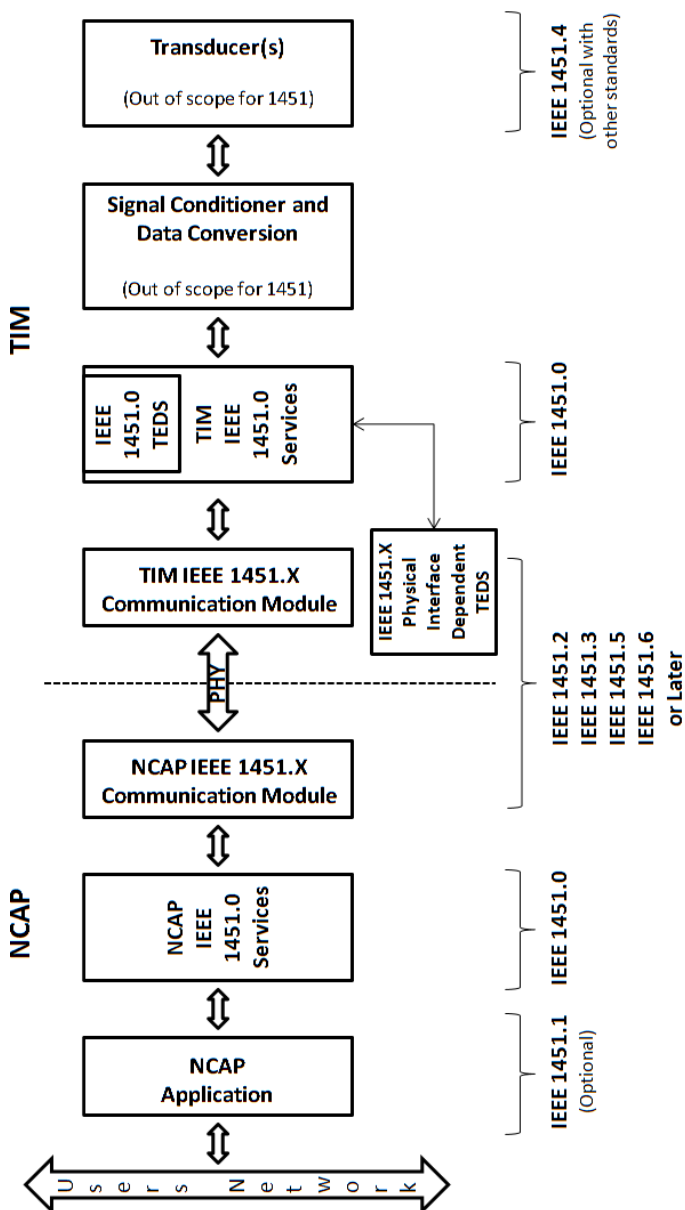


Figura 5: Modelo de referência IEEE 1451.

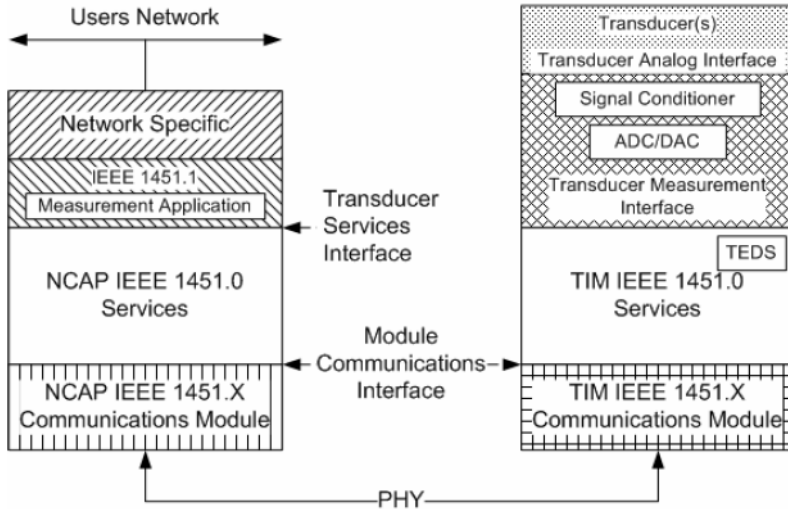


Figura 6: Visão alternativa ao modelo de referência IEEE 1451 (IEEE 1451.0, 2007).

- **IEEE 1451.3:** Apresenta uma interface digital para conectar vários transdutores separados fisicamente. Nela, é definido o formato dos TEDS, a interface elétrica, protocolos de identificação do canal, protocolos *hot swap* e de sincronização de tempo e as funções lógicas de leitura e escrita utilizadas para acessar os TEDS e os dados do transdutor. Assim como a norma IEEE 1451.2, esta não especifica o sistema de condicionamento e conversão de sinal, nem como uma aplicação utiliza as informações dos TEDS. Diferentemente do padrão IEEE 1451.2, IEEE 1451.3 define uma interface transdutor-NCAP e TEDS para utilização de protocolos de comunicação multiponto (IEEE 1451.3, 2004).
- **IEEE 1451.4:** Define um protocolo e uma interface, os quais permitem que transdutores analógicos trafeguem informações digitais com um objeto IEEE 1451. Também, define o formato dos TEDS dos transdutores. A norma não especifica o projeto do transdutor, condicionador de sinal ou o uso específico dos TEDS. Enfim, este padrão define uma interface de modo misto para transdutores analógicos com modos de operação analógico e digital (IEEE 1451.4, 2004).
- **IEEE 1451.5:** Introduz o conceito de WTIM (*Wireless Transducer In-*

terface Module), o qual é conectado através de um módulo de comunicação de rádio aprovado a um módulo de serviço NCAP. Os rádios aprovados pela norma IEEE 1451.5 (*Dot5ApprovedRadio*) são IEEE 802.11, IEEE 802.15.4, IEEE Bluetooth e a tecnologia IEEE ZigBee. WTIM é um módulo que contém um *Dot5ApprovedRadio*, uma unidade de condicionamento de sinal, conversor analógico-digital e/ou digital-analógico e, em muitos casos, os transdutores. Um WTIM pode variar em complexidade, indo de um único sensor ou atuador juntamente com o módulo de rádio, até unidades contendo muitos transdutores mais o rádio. Resumidamente, a especificação desta norma incide sobre os módulos de comunicação que conectam WTIM e NCAP, utilizando os protocolos *Dot5ApprovedRadio* (IEEE 1451.5, 2007).

- **IEEE 1451.6:** Este projeto, ainda em desenvolvimento, estabelece uma rede baseada em CANopen para módulo transdutor multicanal. A norma define o mapeamento de TEDS IEEE 1451 para as entradas do dicionário CANopen, assim como mensagens de comunicação, processamento de dados, parâmetros de configuração e informações de diagnóstico. O padrão adota o perfil do dispositivo CANopen para dispositivos de medição e controladores em malha fechada. Por fim, IEEE 1451.6 define uma camada física CAN intrinsecamente segura (IEEE 1451.6,).
- **IEEE 1451.7:** Propõe formatos de dados para facilitar a comunicação entre sistemas com identificação por radiofrequência (do inglês *Radio-Frequency Identification*, RFID) e etiquetas RFID com transdutores (sensores e atuadores). A padronização define novos formatos de TEDS baseados na família de normas IEEE 1451 e, também, define uma estrutura de comando, assim como especifica os métodos de comunicação com os quais a estrutura de comando foi projetada para ser compatível (IEEE 1451.7, 2010).

3.4 TRANSDUCERCHANNEL

TransducerChannel é o nome dado pela norma para a combinação do transdutor e os componentes de condicionamento e conversão de sinal associados a ele. Cada TransducerChannel está relacionado a um número de identificação contendo 16 bits, definido pelo TIM, servindo como endereço de destino aos comandos enviados pelo NCAP, além de distinguir os TransducerChannels que determinado TIM possui. O TIM, por sua vez, recebe o número de identificação 0, sendo que qualquer comando para este endereço é

encaminhado para o próprio TIM (TORRI, 2008).

3.5 MODOS DE OPERAÇÃO

Cada *TransducerChannel* pode suportar diferentes modos de amostragem e de transmissão (do inglês *sampling and transmission modes*), descritos no padrão IEEE 1451.0. A combinação de ambos forma o modo de operação (do inglês *operating mode*) de um *TransducerChannel*, o qual descreve como será o funcionamento do sensor ou atuador.

O modo de amostragem indica como o *TransducerChannel* deve proceder para realizar a recuperação de dados, caso o mesmo for um sensor, ou como os dados de entrada devem ser utilizados, quando o transdutor é um atuador. Este modo pode, também, definir o intervalo, em tempo ou outra magnitude, no qual os dados devem ser recuperados/utilizados.

O modo de transmissão determina como a informação é transferida entre NCAP e TIM. O modo, ou modos, de operação que um *TransducerChannel* pode operar é indicado em atributos no *TransducerChannel TEDS* de cada transdutor (TORRI, 2008).

A norma especifica cinco possíveis modos de amostragem para o funcionamento dos transdutores. São eles: *trigger-initiated*, *free-running without pre-trigger*, *free-running with pre-trigger (with or without buffers enabled)*, *continuous sampling* e *immediate operation*. Enquanto que os três modos de transmissão possíveis são: *only when commanded*, *streaming when a buffer is full* e *streaming at a fixed interval*.

A escolha do modo de amostragem *continuous sampling*, em combinação com o modo de transmissão *streaming when a buffer is full* ou *streaming at a fixed interval*, forma um modo de operação conhecido como *streaming operation*. Um sensor executando neste modo de operação deve adquirir os dados e transmiti-los ao NCAP assim que seu conjunto de leituras tenha sido preenchido ou em intervalos periódicos, sem a necessidade de receber comandos adicionais de polling vindos do NCAP. Assim, o modo *streaming operation* permite que as leituras feitas por um sensor sejam enviadas para o NCAP de forma autônoma e automática (TORRI, 2008).

3.6 COMUNICAÇÃO SEM FIO ENTRE TIM E NCAP

O padrão IEEE 1451.0 descreve três estruturas de mensagens, as quais são enviadas entre NCAP e TIM através da interface de comunicação. Estas mensagens são chamadas de comando, resposta e mensagem iniciada pelo

TIM, e devem ser implementadas independentemente do meio de comunicação físico utilizado. Ao lidar com sensores sem fio, implica-se na conexão entre NCAP e WTIM através de radiofrequência, descrita pela norma IEEE 1451.5. Neste caso, o WTIM conterá um módulo de rádio para comunicação e, por sua vez, o NCAP deve possuir um rádio semelhante que completará o link de comunicação entre ambos.

Song e Lee descrevem uma implementação dos padrões IEEE 1451.0 e 1451.5, utilizando a linguagem de programação Java. Este sistema consiste em dois nós, NCAP e WTIM, os quais se comunicam usando as interfaces padrões, através do módulo de comunicação sem fio IEEE 802.11. Três exemplos de implementação das interfaces são discutidos por eles. O primeiro exemplo foca no anúncio e descoberta do nó sem fio, utilizando o modelo publicar-assinar (do inglês *publish-subscribe*). O segundo concentra-se em requisição-resposta de dados do sensor, utilizando o modelo cliente-servidor (do inglês *client-server*). E, por fim, o terceiro exemplo foca na requisição-resposta de TEDS presentes no sensor, através, também, do modelo cliente-servidor (SONG; LEE, 2006) (LEE; SONG, 2007).

4 PROTOCOLO SIP

SIP (*Session Initiation Protocol*) é um protocolo presente na camada de sessão responsável por estabelecer, modificar e finalizar sessões para comunicação multimídia entre usuários, tal como chamada de voz e/ou vídeo (ROSENBERG et al., 2002). Este protocolo é utilizado para estabelecer chamadas através de redes via protocolo IP (*Internet Protocol*) e normalmente utiliza os protocolos UDP (*User Datagram Protocol*) ou TCP (*Transmission Control Protocol*) para a camada de transporte, porém, outros protocolos também podem ser utilizados, por exemplo, SCTP (*Stream Control Transmission Protocol*) e TLS (*Transport Layer Security*). A figura 7 apresenta o modelo de camadas do protocolo SIP, incluindo alguns outros protocolos conhecidos, os quais estão relacionados ou são comumente utilizados em conjunto com o mesmo.

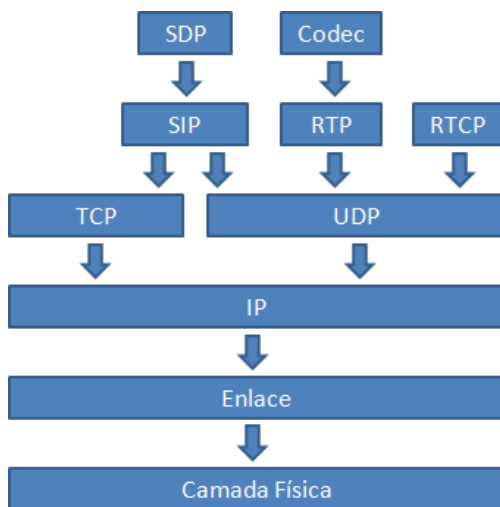


Figura 7: Modelo de camadas do protocolo SIP.

SDP (*Session Description Protocol*), definido pelo RFC 4566 (HANDLEY; JACOBSON; PERKINS, 2006), representa um protocolo descrito em formato texto e responsável pela descrição dos parâmetros de uma sessão multimídia. Algumas das informações tratadas pelo protocolo são: endereço IP do usuário, número da porta utilizada para o transporte, tipo de mídia trafegada (como áudio e vídeo), codecs utilizados, entre outras. Requisições SIP as quais têm por finalidade estabelecer uma sessão enviam em seu corpo da-

dos utilizando o protocolo SDP, ou seja, este é encapsulado dentro de mensagens SIP.

O RFC 3550 descreve o protocolo RTP (*Real-Time Transport Protocol*) (SCHULZRINNE et al., 2003). O mesmo prove funções de transporte adaptáveis para aplicações as quais transmitem áudio e/ou vídeo em tempo real. Um pacote RTP inclui informações tais como a identificação do tipo do payload (codec), número de sequência, tempo de entrega de cada fragmento (*timestamp*), entre outras. Normalmente, utiliza-se o protocolo UDP para a camada de transporte e RTP suporta a transferência de dados a múltiplos destinatários, usando a distribuição multicast, além do serviço unicast tradicional. Já o RTCP (*Real-Time Transport Control Protocol*) representa um protocolo também definido pelo RFC 3550, permitindo que participantes de uma sessão RTP enviem relatórios e estatísticas a outros usuários relatando a qualidade da mídia. Assim, é possível executar diagnósticos e modificar as taxas de transmissão, por exemplo.

4.1 ENTIDADES

O protocolo SIP define alguns conceitos importantes ao entendimento de seu funcionamento, tal como a definição de cliente e servidor. Um cliente é um elemento de rede que envia requisições SIP e recebe respostas. Clientes podem ou não interagir diretamente com um usuário humano e exemplos são *user agent clients* e *proxies*. Já um servidor é um elemento de rede que recebe requisições a fim de servi-las, devendo retornar respostas. Exemplos são *user agent servers*, *proxies*, *redirect servers* e *registrar servers*.

Estas entidades citadas tratam-se de conceitos fundamentais ao protocolo. Sendo assim, serão apresentados a seguir.

- **User Agent (UA):** É uma entidade lógica que pode atuar tanto como *user agent client* (UAC) quanto como *user agent server* (UAS). Um UAC é uma entidade que cria novas requisições e envia-as. O papel de um UAC dura somente o tempo necessário para o término da transação em questão, ou seja, caso seja recebida uma requisição posteriormente, o mesmo agente assumirá o papel de um UAS para processar àquela nova transação. Já um UAS é uma entidade lógica que gera uma resposta a uma requisição SIP entrante. A resposta pode aceitar, rejeitar ou redirecionar a requisição. E, assim como o UAC, seu papel dura somente o tempo da transação.
- **Proxy Server:** Representa uma entidade intermediária, a qual atua tanto como servidor quanto como cliente, sendo seu propósito enviar

requisições em nome de outros dispositivos. Um *proxy server* tem como papel primário fazer o roteamento de mensagens, garantindo que uma requisição seja enviada para uma entidade mais próxima ao usuário alvo. O mesmo interpreta e, se necessário, reescreve partes específicas da mensagem antes de enviá-la adiante. *Proxies* são também úteis para políticas de verificação, conferindo se é permitido ao usuário realizar a chamada requisitada. Um *proxy* pode ser classificado como *stateful* ou *stateless*. *Stateful proxy* é uma entidade que mantém as máquinas de estado das transações clientes e servidoras durante o processo de uma requisição, enquanto que o *stateless proxy* não registra as informações relacionadas às transações, somente enviando adiante cada requisição e resposta que o mesmo recebe.

- **Redirect Server:** Esta entidade nada mais é do que um UAS que gera respostas indicando redirecionamento a requisições recebidas. Estas respostas buscam redirecionar o cliente, fazendo-o contactar um conjunto alternativo de URIs (*Uniform Resource Identifier*), com a finalidade de alcançar seu destino.
- **Registrar Server:** Representa um servidor o qual aceita requisições SIP chamadas *REGISTER*, registrando as informações contidas nestas mensagens em um serviço de localização, redirecionando as mensagens posteriormente recebidas ao usuário apropriado.

4.2 FORMATO DAS MENSAGENS

Requisições e respostas SIP possuem campos de cabeçalho, do inglês *header fields*. Estes cabeçalhos fornecem informações sobre a mensagem e sobre o corpo que a mesma contém. Alguns podem ser utilizados tanto em requisições como em respostas, porém alguns são específicos para somente um dos tipos. O cabeçalho consiste em um nome identificador, seguido por dois pontos (:) e finalizando com um valor para aquele campo.

O protocolo SIP possui 44 campos de cabeçalho descritos em sua norma. Uma breve descrição de cada um deles é apresentada na tabela 3:

Tabela 3: Campos de cabeçalho SIP.

Nome	Descrição
<i>Accept</i>	Indica os tipos de mídia aceitas para o corpo da mensagem

Tabela 3: Campos de cabeçalho SIP (continuação).

Nome	Descrição
<i>Accept-Encoding</i>	Especifica os esquemas de codificação aceitos
<i>Accept-Language</i>	Especifica preferências de idiomas para mensagens e campos de cabeçalho
<i>Alert-Info</i>	Apresenta um <i>ring tone</i> ou <i>ringback tone</i> para o UAS ou UAC
<i>Allow</i>	Indica os métodos suportados pelo <i>user agent</i> ou <i>proxy server</i>
<i>Authentication-Info</i>	Inserido em respostas quando necessário realizar autenticação mútua
<i>Authorization</i>	Transmite as credenciais de um UA
<i>Call-ID</i>	Identifica unicamente uma chamada entre dois <i>user agents</i>
<i>Call-Info</i>	Provê informação adicional sobre o originador da mensagem
<i>Contact</i>	Fornece um endereço para contato direto
<i>Content-Disposition</i>	Informa a função do corpo da mensagem SIP
<i>Content-Encoding</i>	Lista os esquemas de codificação aplicados ao corpo da mensagem
<i>Content-Language</i>	Indica o idioma do corpo da mensagem
<i>Content-Length</i>	Fornece o número de octetos contidos no corpo da mensagem SIP
<i>Content-Type</i>	Provê o tipo de mídia contido no corpo da mensagem
<i>CSeq</i>	Apresenta o número de sequência da mensagem
<i>Date</i>	Informa a data de quando a requisição ou resposta foi enviada
<i>Error-Info</i>	Provê informação adicional referente a uma resposta indicando falha
<i>Expires</i>	Indica o intervalo de tempo que o conteúdo de uma mensagem é válido
<i>From</i>	Informa o usuário de origem da requisição
<i>In-Reply-To</i>	Enumera os Call-IDs que a chamada em questão referencia ou retorna
<i>Max-Forwards</i>	Indica o número máximo de pulos que uma mensagem pode executar
<i>MIME-Version</i>	Indica a versão do protocolo MIME usado
<i>Min-Expires</i>	Tempo mínimo de expiração suportado por um <i>registrar server</i>

Tabela 3: Campos de cabeçalho SIP (continuação).

Nome	Descrição
<i>Organization</i>	Apresenta a organização do originador da mensagem
<i>Priority</i>	Indica a urgência de uma requisição do ponto de vista do UAC
<i>Proxy-Authenticate</i>	Contém um desafio de autenticação partindo de um <i>proxy server</i>
<i>Proxy-Authorization</i>	Identifica um usuário perante a um <i>proxy</i> o qual requisitou autenticação
<i>Proxy-Require</i>	Lista as características que um <i>user agent</i> requer que o <i>proxy</i> suporte
<i>Record-Route</i>	Força o roteamento através de um <i>proxy</i>
<i>Reply-To</i>	Especifica um URI para a resposta à requisição
<i>Require</i>	Lista opções as quais um UAC requer que o UAS suporte
<i>Retry-After</i>	Tempo em que um serviço poderá estar disponível novamente
<i>Route</i>	Provê informação de roteamento para requisições
<i>Server</i>	Informa o software usado pelo UAS para tratar os métodos
<i>Subject</i>	Provê indicação da natureza de uma chamada, ou seja, o seu objetivo
<i>Supported</i>	Apresenta uma ou mais extensões implementadas pelo UA ou servidor
<i>Timestamp</i>	Descreve o momento exato do envio de uma requisição
<i>To</i>	Informa o usuário de destino da presente requisição
<i>Unsupported</i>	Apresenta características não suportadas pelo UAS
<i>User-Agent</i>	Informa o software utilizado pelo UAC para originar requisições
<i>Via</i>	Registra a rota tomada por uma requisição
<i>Warning</i>	Carrega informação adicional sobre o estado de uma resposta
<i>WWW-Authenticate</i>	Contém um desafio de autenticação

4.3 REQUISIÇÕES

O protocolo SIP possui seis requisições especificadas em sua norma (ROSENBERG et al., 2002): *INVITE*, *ACK*, *BYE*, *CANCEL*, *REGISTER* e *OPTIONS*. Outros documentos RFC (*Request for Comments*) separados descrevem algumas mensagens opcionais, possibilitando a extensão do protocolo, tais como: *REFER* (SPARKS, 2003), *MESSAGE* (CAMPBELL et al., 2002), *UPDATE* (ROSENBERG, 2002), *INFO* (DONOVAN, 2000), *PRACK* (ROSENBERG; SCHULZRINNE, 2002b), *SUBSCRIBE* e *NOTIFY* (ROACH, 2002). Requisições também são chamadas de métodos e as mesmas requisitam uma ação específica a ser tomada pelo *user agent* ou servidor destinado. A seguir serão apresentados os métodos SIP originais, ou seja, os especificados no RFC 3261 (ROSENBERG et al., 2002), assim como algumas outras requisições necessárias para o decorrer deste trabalho.

4.3.1 INVITE

O método *INVITE* é utilizado a fim de estabelecer sessões multimídia entre UAs. Esta requisição normalmente possui dados adicionais no corpo da mensagem, contendo informações sobre as mídias aceitas pela parte chamadora, por exemplo, áudio e/ou vídeo e os respectivos codecs suportados. A inicialização de uma sessão multimídia consiste em um *three-way handshake*, ou seja, para que seja considerada estabelecida, necessita-se que sejam transferidas três mensagens entre o UAC e o UAS: uma requisição *INVITE*, uma resposta contendo a informação 200 OK e um método *ACK* finalizando o processo. Uma requisição *INVITE* executada com sucesso estabelece um diálogo entre os dois *user agents*, o qual continuará até que uma mensagem *BYE* seja enviada por uma das partes envolvidas na chamada, finalizando a sessão.

Após a inicialização de uma sessão, novas requisições *INVITEs* podem ser enviadas, sendo que as mesmas solicitam a modificação da sessão. Esta mensagem, chamada *re-INVITE*, transmite uma nova descrição da sessão, contendo novos parâmetros, tais como o número da porta em que os fluxos multimídia são recebidos, ou até mesmo modificando os tipos de mídias trafegados.

O protocolo SIP trata somente os convites e as respostas enviadas pelos usuários para o estabelecimento do diálogo. As informações relativas às sessões são tratadas pelo protocolo de descrição de sessão utilizado, normalmente o SDP (ROSENBERG; SCHULZRINNE, 2002a) (HANDLEY; JACOBSON; PERKINS, 2006). Também, SIP é habitualmente utilizado em

conjunto com o protocolo RTP, o qual é empregado em aplicações de tempo real para o tratamento de fluxos de áudio e vídeo. Um exemplo da requisição *INVITE* é apresentado a seguir, enquanto que a figura 8 apresenta o fluxo de mensagens para que uma sessão SIP seja inicializada, conforme descrito pela norma.

```
INVITE sip:user2@host2.com SIP/2.0
Via: SIP/2.0/UDP pc.host1.com;branch=z9hG4bKcb6s8k
From: User1 <sip:user1@host1.com>;tag=25c6e14
To: User2 <sip:user2@host2.com>
Call-ID: b93dc567e0891
CSeq: 12592 INVITE
Contact: <sip:user1@pc.host1.com>
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: 142
```

(SDP)

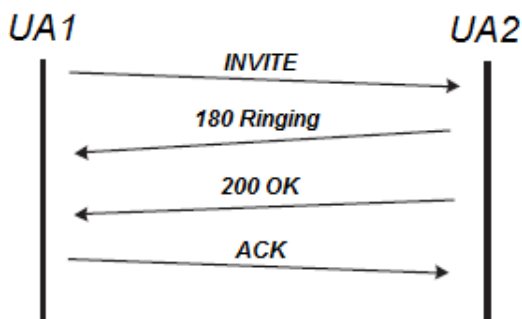


Figura 8: Estabelecimento de uma sessão SIP.

4.3.2 ACK

Uma mensagem *ACK* é utilizada para reconhecer a recepção de uma resposta final às requisições *INVITE*. Assim, o *three-way handshake* é completado quando o UAC, o qual enviou o primeiro *INVITE*, emite uma requisição *ACK* ao receber uma resposta final ao pedido de inicialização da sessão.

Um método *ACK* pode conter um corpo na mensagem com uma descrição de sessão caso a requisição *INVITE* não possuía, caso contrário, a mensagem de reconhecimento não poderá transmitir esta informação. Ou seja, a

requisição *ACK* não pode ser utilizada para modificar uma descrição de mídia a qual já foi previamente enviada no convite inicial. Para este propósito, deve-se utilizar uma mensagem *re-INVITE* após o estabelecimento da sessão.

A requisição *ACK* juntamente com a mensagem *INVITE*, utilizadas para inicializar uma sessão, são apresentadas na figura 8. Abaixo possui um exemplo do método *ACK*.

```
ACK sip:user2@host2.com SIP/2.0
Via: SIP/2.0/UDP pc.host1.com;branch=z9hG4bKaj7hd89
From: User1 <sip:user1@host1.com>;tag=12ce4377
To: User2 <sip:user2@host2.com>;tag=46b9f3de
Call-ID: 434fa2210ded32
CSeq: 33421 ACK
Max-Forwards: 70
Content-Length: 0
```

4.3.3 BYE

O método *BYE* é utilizado por um UA a fim de abandonar uma sessão multimídia estabelecida. Em sessões entre duas partes, o abandono por um dos lados implica no término da sessão. Já em cenários multicast, uma mensagem *BYE* indica, apenas, que um dos participantes deixou a conferência, não afetando a sessão como um todo (CAMARILLO, 2001).

Uma requisição *BYE* não deve ser utilizada para cancelar *INVITES* pendentes, ou seja, os quais ainda não tenham completado o *three-way handshake*. Para isto, uma requisição *CANCEL* deve ser usada. Um exemplo de uma mensagem *BYE* é fornecido a seguir e a figura 9 apresenta o término de uma sessão SIP.

```
BYE sip:user1@host1.com SIP/2.0
Via: SIP/2.0/UDP host2.com;branch=z9hG4bKab4d56f0
From: User2 <sip:user2@host2.com>;tag=bc54cdf
To: User1 <sip:user1@host1.com>;tag=217cde44
Call-ID: 7734sad2a133
CSeq: 34566 BYE
Max-Forwards: 70
Content-Length: 0
```

4.3.4 CANCEL

O método *CANCEL* é utilizado para cancelar transações pendentes, tais como buscas e tentativas de chamadas. O mesmo é utilizado somente enquanto nenhuma resposta final tenha sido retornada. Após o recebimento

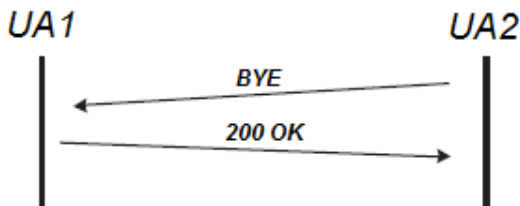


Figura 9: Finalização de uma sessão SIP.

de uma resposta, a mensagem *CANCEL* não possui efeito na transação, sendo necessário utilizar a requisição *BYE* para o cancelamento da chamada.

Requisições *CANCEL* também são úteis quando *forking proxies* estão no caminho do estabelecimento de uma sessão. *Forking proxies* emitem mensagens *INVITE* para um grupo de destinatários em paralelo, ao invés de somente um, sendo que após o recebimento de uma resposta final a partir de qualquer um dos usuários destinos, as demais tentativas de buscas iniciadas devem ser canceladas, o que é feito através do método *CANCEL*.

A figura 10 demonstra o cancelamento de uma tentativa de estabelecimento de sessão SIP. A seguir é apresentado um exemplo de uma mensagem *CANCEL*.

```

CANCEL sip:user2@host2.com SIP/2.0
Via: SIP/2.0/UDP host1.com;branch=z9hG4bK433fs23bb
From: User1 <sip:user1@host1.com>;tag=2ab455dsf3
To: User2 <sip:user2@host2.com>
Call-ID: acd22b3467sda
CSeq: 22346 CANCEL
Max-Forwards:70
Content-Length: 0
  
```

4.3.5 REGISTER

Usuários enviam requisições *REGISTER* com a finalidade de informar um *registrar server* sobre suas atuais localizações. O processo de registro não é requerido para habilitar um UA a usar um *proxy server* para chamadas de saída. Porém, é necessário para que o *user agent* receba chamadas entrantes de *proxies* que servem àquele domínio, a menos que mecanismos diferentes do descrito pelo padrão SIP sejam utilizados pelo serviço de localização para popular os URIs e os endereços de contato dos usuários finais (JOHNSTON, 2003).

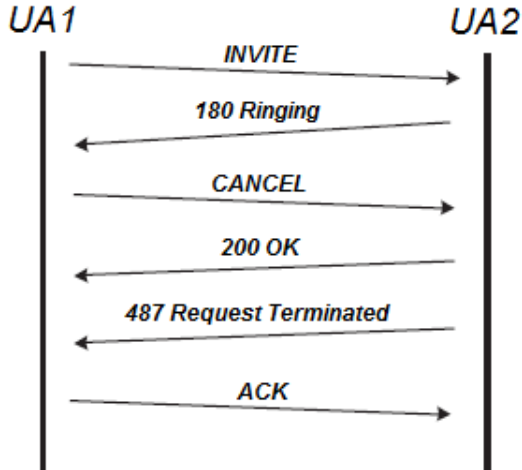


Figura 10: Inicialização de uma sessão SIP sendo cancelada.

A presença do campo de cabeçalho *Expires* define um tempo, em segundos, em que o presente registro será válido. Caso este campo não seja especificado, o valor padrão para que o registro expire é de 1 hora. Também, um UA pode ser registrado como estando em várias localizações ao mesmo tempo, indicando que o servidor deveria procurar pelo usuário em todos os locais registrados até que o mesmo seja encontrado. Isto é feito através de *forking proxies*.

Um exemplo de uma mensagem *REGISTER* é fornecido abaixo, enquanto que a figura 11 apresenta o registro de um usuário SIP.

```

REGISTER sip:registrar.com SIP/2.0
Via: SIP/2.0/UDP host.com:5060;branch=z9hG4bKffa452bc7
From: User <sip:user@host.com>;tag=22fg467s2
To: User <sip:user@host.com>
Call-ID: 23ad45ffab78
CSeq: 23675 REGISTER
Contact: <sip:user@169.254.1.3>
Max-Forwards: 70
Expires: 3600
Content-Length: 0
  
```




Figura 11: Registro de um usuário SIP.

4.3.6 OPTIONS

O método *OPTIONS* é utilizado para consultar um *user agent* ou servidor sobre suas capacidades e descobrir sua disponibilidade atual. Algumas capacidades descobertas pela referida requisição são os métodos e protocolos de descrição de sessão os quais são suportados, assim como quais esquemas de compressão são reconhecidos para o corpo da mensagem, permitindo o envio de informações comprimidas, economizando a largura de banda da rede.

Um usuário SIP consultando sobre as capacidades de outro é apresentado na figura 12. Um exemplo de uma mensagem *OPTIONS* é fornecido abaixo.

```

OPTIONS sip:user2@host2.com SIP/2.0
Via: SIP/2.0/UDP pc.host1.com;branch=z9hG4bKah88fsb65
From: User1 <sip:user1@host1.com>;tag=23sfd43ad
To: User2 <sip:user2@host2.com>
Call-ID: fd456aac56ss90
CSeq: 43102 OPTIONS
Contact: <sip:user1@pc.host1.com>
Max-Forwards: 70
Accept: application/sdp
Content-Length: 0
  
```

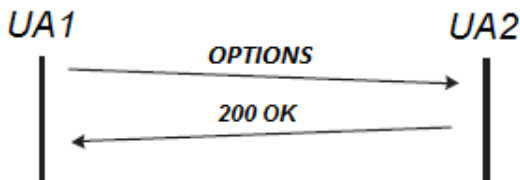


Figura 12: Um usuário SIP consultando sobre as capacidades de outro.

4.3.7 SUBSCRIBE e NOTIFY

O documento “Session Initiation Protocol (SIP)-Specific Event Notification” (ROACH, 2002) descreve uma extensão ao protocolo SIP, definindo dois novos métodos, *SUBSCRIBE* e *NOTIFY*. Os mesmos permitem que usuários SIP requisitem notificações indicando a ocorrência de determinados eventos em usuários remotos.

A requisição *SUBSCRIBE* é utilizada por um *user agent* para estabelecer uma assinatura com o propósito de receber notificações, via método *NOTIFY*, sobre um evento particular. Uma assinatura realizada com sucesso estabelece um diálogo entre UAC e UAS, o qual permanece ativo pelo tempo indicado no campo de cabeçalho *Expires*. Quando este tempo expira, a assinatura é automaticamente finalizada, porém, novas mensagens *SUBSCRIBE* podem ser enviadas a fim de atualizar o tempo restante. O mesmo método, porém com o cabeçalho *Expires* indicando o valor 0, deve ser enviado para terminar uma assinatura e o diálogo associado (JOHNSTON, 2003).

Para que um *user agent* relate a ocorrência de um particular evento, é utilizado o método *NOTIFY*. Esta requisição é sempre enviada dentro de um diálogo, tal como o criado pelo método *SUBSCRIBE*.

A funcionalidade descrita por ambas as mensagens, notificação assíncronas de eventos, pode ser utilizada em uma diversidade de situações onde a cooperação entre usuários SIP é requerida, por exemplo, serviços de *callback*, listas de presença, indicações de mensagens em espera, entre outros. Devido às extensas possibilidades de uso, a norma correspondente a estes métodos representa um framework extensível, o qual deve ser seguido por outros RFCs, definindo aplicações específicas. Um exemplo é o RFC 3856 (ROSENBERG, 2004), utilizado para notificação de presença, permitindo reconhecer o estado de conectividade de um usuário remoto, ou seja, se o mesmo está *on-line* ou *off-line*.

A figura 13 demonstra o funcionamento das mensagens *SUBSCRIBE* e *NOTIFY*, onde a primeira requisita notificações com relação ao estado do usuário remoto, enquanto que a última fornece as informações solicitadas. Abaixo são apresentados exemplos de codificação de ambas as mensagens. Neste exemplo, o corpo do método *NOTIFY* contém o formato PIDF (*Presence Information Data Format*) (SUGANO et al., 2004), o qual indica o estado de conectividade do usuário.

```
SUBSCRIBE sip:user2@host2.com SIP/2.0
Via: SIP/2.0/UDP host1.com;branch=z9hG4bKadf43gh6
From: User1 <sip:user1@host1.com>;tag=25fdsa35
To: User2 <sip:user2@host2.com>
Call-ID: afd367cba98s
```

```

CSeq: 34226 SUBSCRIBE
Max-Forwards: 70
Event: presence
Accept: application/pidf+xml
Contact: <sip:user1@pc.host1.com>
Expires: 3600
Content-Length: 0

```

```

NOTIFY sip:user1@host1.com SIP/2.0
Via: SIP/2.0/UDP host2.com;branch=z9hG4bKsa982k22
From: User2 <sip:user2@host2.com>;tag=dd346fs9
To: User1 <sip:user1@host1.com>;tag=25fdsa35
Call-ID: afd367cba98s
CSeq: 5983 NOTIFY
Max-Forwards: 70
Event: presence
Subscription-State: active;expires=3599
Contact: sip:user2@pc.host2.com
Content-Type: application/pidf+xml
Content-Length: 290

```

(PIDF)

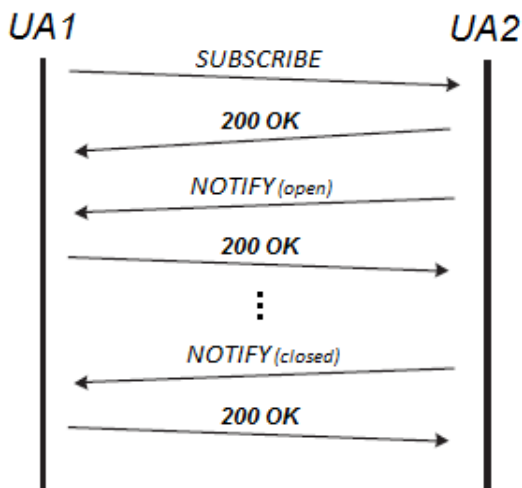


Figura 13: Inscrição e notificação de eventos utilizando SIP.

4.3.8 MESSAGE

A requisição *MESSAGE* corresponde a uma extensão ao protocolo SIP, permitindo a transferência de mensagens instantâneas (do inglês *Instant Messaging*, IM). IM refere-se ao envio de mensagens em tempo real entre usuários, ou seja, as requisições devem ser transferidas de forma a manter uma conversa interativa entre os participantes (CAMPBELL et al., 2002).

Métodos *MESSAGES* transportam conteúdo em seu corpo no formato MIME (*Multipurpose Internet Mail Extensions*), por exemplo, *plain/text*, *message/cpim*, *text/html*, entre outros. Também, esta mensagem não estabelece um diálogo e, normalmente, são enviadas de forma independente uma das outras, tais como mensagens de um pager.

A figura 14 apresenta um exemplo de envio do método *MESSAGE* e abaixo é fornecida a codificação da mensagem.

```
MESSAGE sip:user2@host2.com SIP/2.0
Via: SIP/2.0/UDP host1.com;branch=z9hG4bKsd59acb2
From: sip:user1@host1.com;tag=682ad9bb2
To: sip:user2@host2.com
Call-ID: d10sa892fh1
CSeq: 3411 MESSAGE
Max-Forwards: 70
Content-Type: text/plain
Content-Length: 17
```

Texto de exemplo.

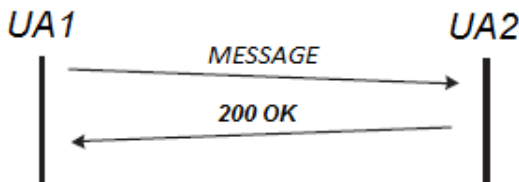


Figura 14: Envio de uma mensagem instantânea utilizando SIP.

4.4 RESPOSTAS

Uma resposta SIP é uma mensagem gerada por um *user agent server* ou um *SIP server* como resposta a uma requisição gerada por um *user agent*.

client. Cada resposta possui um código que indica o estado da transação. Estes códigos de estado são valores inteiros na faixa de 100 a 699 e são agrupados em seis classes, como mostrado na tabela 4.

Tabela 4: Classes das respostas SIP.

Classe	Descrição
1xx	Informacional
2xx	Sucesso
3xx	Redirecionamento
4xx	Erro no Cliente
5xx	Erro no Servidor
6xx	Falha Global

Uma resposta com o código entre 100 e 199 é considerada provisória, enquanto que respostas na faixa de 200 até 699 são finais. Uma transação SIP entre um cliente e um servidor compreende uma requisição partindo do cliente, podendo haver uma ou mais respostas provisórias, dependendo do tipo da requisição em andamento, e finalizando com uma resposta final enviada pelo servidor.

Uma resposta SIP possui, além do código de estado, uma frase razão. Esta frase serve somente para a leitura humana, transmitindo informação em formato texto para facilitar a identificação da resposta por uma pessoa, sendo provavelmente apresentada ao usuário destino. Um computador processando uma resposta normalmente ignora esta frase, pois o código numérico possui informação suficiente para que o mesmo analise a mensagem.

A tabela 5 contém os códigos de estado atualmente definidos, juntamente com suas respectivas frases padrões.

Tabela 5: Códigos das respostas SIP.

Código	Frase Razão
100	Trying
180	Ringin
181	Call Is Being Forwarded
182	Queued
183	Session Progress
200	OK
202	Accepted

Tabela 5: Códigos das respostas SIP (continuação).

Código	Frase Razão
300	Multiple Choices
301	Moved Permanently
302	Moved Temporarily
305	Use Proxy
380	Alternative Service
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Time-out
409	Conflict
410	Gone
411	Length Required
413	Request Entity Too Large
414	Request-URI Too Long
415	Unsupported Media Type
416	Unsupported URI Scheme
420	Bad Extension
421	Extension Required
423	Interval Too Brief
480	Temporarily Unavailable
481	Call/Transaction Does Not Exist
482	Loop Detected
483	Too Many Hops
484	Address Incomplete
485	Ambiguous
486	Busy Here
487	Request Terminated
488	Not Acceptable Here
491	Request Pending
493	Undecipherable
500	Server Internal Error
501	Not Implemented
502	Bad Gateway

Tabela 5: Códigos das respostas SIP (continuação).

Código	Frase Razão
503	Service Unavailable
504	Server Time-out
505	Version Not Supported
513	Message Too Large
600	Busy Everywhere
603	Decline
604	Does Not Exist Anywhere
606	Not Acceptable

4.5 SIP PARA SISTEMAS EMBARCADOS

SIP vem sendo largamente utilizado em sistemas embarcados, como celulares, PDAs e web câmeras (LAKAY; AGBINYA, 2005). Entretanto, o protocolo carece em desempenho, por exemplo, em relação ao tamanho do código. Sendo assim, o mesmo não se torna atrativo em sistemas embarcados com restrição de recursos, tais como eletrodomésticos e câmeras de segurança de baixo custo.

Alguns dos trabalhos desenvolvidos em cima do protocolo SIP exploram *home gateways* para o uso em dispositivos domésticos. *Real-Time Visitor Communication Service* (RVCS) é uma arquitetura baseada no sistema de *home gateway* e no protocolo SIP, na qual o dono de uma casa pode monitorar visitantes localizados em frente a sua porta utilizando dispositivos conectados à Internet, tais como PCs e PDAs, suportando a mobilidade do usuário. Primeiramente, o dono da residência deve registrar seu dispositivo, móvel ou não, ao *home gateway*, então, quando um visitante toca a campainha da porta, o gateway recebe o sinal e inicia uma sessão SIP com o dispositivo registrado previamente. Assim, dados de áudio e vídeo são trocados entre eles até que a conversação seja finalizada. Além disso, o morador pode abrir a porta de sua casa remotamente, através de seu dispositivo (OH; PAIK; PARK, 2006).

SIP Context-Aware Gateway (SCAG) representa um gateway inteligente implantado entre a rede doméstica e a Internet, permitindo ao proprietário da casa usar dispositivos SIP de sua preferência para comunicar e postar suas informações de contexto (situação do ambiente atual) para aparelhos domésticos ou para outras pessoas, conforme necessário. Assim, aplicações para casas inteligentes (do inglês *smart homes*) podem oferecer serviços que possam ser adaptados às situações dinâmicas do usuário, tal como a

localização do mesmo, e que deem tratamento especial às suas preferências (CHENG; CHEN; TSENG, 2006).

Kwak propõe um projeto de um sistema de serviço ubíquo (onipresente), provendo mobilidade e interface uniforme ao usuário (KWAK, 2007). Para isto, foram utilizados os protocolos SIP e X.10, onde o primeiro permite que sessões de serviço sejam preservadas mesmo quando o usuário move-se de um ambiente de computação para outro, enquanto que o segundo representa um protocolo de comunicação para o controle remoto de dispositivos elétricos. Quando um usuário deseja entrar em um *SmartSpace* (cenário que consiste de serviços inteligentes acessíveis a usuários móveis via dispositivos de bolso conectados sobre enlaces sem fio de curta distância), ele registra seu dispositivo em um gerenciador. Assim, o dispositivo portátil recebe uma lista atualizada com os dispositivos que existem no *SmartSpace* e, após escolher o serviço desejado e o aparelho de destino, o usuário é capaz de conectar-se a ele através de uma interface uniforme, baseada em operações de eventos de toque. Então, o serviço solicitado pode ser oferecido ao usuário.

A habilidade de processar mensagens SIP rapidamente é crítica para o desempenho de dispositivos eletrônicos de consumo na rede doméstica. Levando em conta que o parse de mensagens inteiras representa o gargalo de desempenho de servidores SIP e sabendo que o *home gateway* e os dispositivos eletrônicos de consumo executam funções diferentes e, assim, necessitam de informações diferentes vindas das mensagens SIP, *Demand-Driven Parsing Method* (DPM) somente executa o parse de algumas partes da mensagem SIP. DPM é descrito por documentos XML (*Extensible Markup Language*) configurados no servidor SIP, os quais são carregados no momento em que o sistema é iniciado. Ao receber mensagens SIP, os campos de cabeçalho pré-descritos no documento XML são analisados pelo parser do servidor, enquanto que os outros não mencionados no documento são ignorados. Como os campos de cabeçalho e seus valores são analisados sob demanda, a maior parte das mensagens SIP permanece não analisada, permitindo uma economia no tempo de processamento comparado à análise completa (LIAO et al., 2009).

5 TRABALHOS RELACIONADOS

A norma IEEE 1451 especifica duas maneiras possíveis para acessar transdutores através de uma rede, ambas opcionais. A primeira delas utiliza o padrão IEEE 1451.1 (IEEE 1451.1, 2000) e a segunda é conhecida como IEEE 1451.0 HTTP (*Hypertext Transfer Protocol*).

O padrão IEEE 1451.1 define uma interface para conectar NCAPs a redes de controle, através do desenvolvimento de um modelo de objeto comum para sensores e atuadores inteligentes (IEEE 1451.1, 2000). O mesmo visa fornecer duas camadas de abstrações: a primeira relacionada à rede e a segunda, ao transdutor. O modelo de objeto fornecido provê um framework extensível para a implementação de software para sistemas de controle e medição distribuídos. A figura 15 apresenta a arquitetura de camadas do framework orientado a objetos para o padrão IEEE 1451.1. A camada IEEE 1451.1 da figura foca na hierarquia de classe do padrão referido, consistindo de um modelo neutro e um ou mais modelos baseados em middleware. O modelo neutro, o qual é independente de rede e middleware, permite que desenvolvedores de aplicações estendam-no, visando implementar seus sistemas específicos. Este modelo consiste de duas partes: modelo de dados e modelo de objetos. O modelo de dados especifica os tipos e formatos das informações comunicadas entre as interfaces do padrão IEEE 1451, tanto para comunicação local quanto remota. Alguns exemplos de tipos de dados são *boolean*, *integer*, *string*, entre outros. O modelo de objetos especifica os tipos de componentes de software utilizados para projetar e implementar os sistemas da camada de aplicação, por exemplo, contendo as classes responsáveis pela comunicação cliente-servidor e publicar-assinar. Por último, a camada de aplicação foca no projeto do sistema final, o qual é formado pela composição ou agregação dos objetos da camada inferior. Suas classes herdam das presentes no modelo baseado em middleware do framework.

A segunda forma de acessar transdutores é através da API IEEE 1451.0 HTTP (IEEE 1451.0, 2007), focando principalmente no acesso a dados e TEDS dos transdutores, usando o protocolo HTTP 1.1. O HTTP, presente na camada de aplicação do modelo OSI, é utilizado para transmitir informações na *World Wide Web*. Este protocolo é baseado no conceito de conexão cliente-servidor, onde dois processadores podem se comunicar através de uma conexão TCP/IP. O lado servidor é responsável por aguardar requisições HTTP em uma determinada porta, enquanto que a parte cliente tem a missão de estabelecer uma conexão via *socket* com o servidor, transmitir uma requisição e esperar por uma resposta originada pelo lado remoto. A API IEEE 1451.0 HTTP descreve a implementação somente do lado servidor, represen-

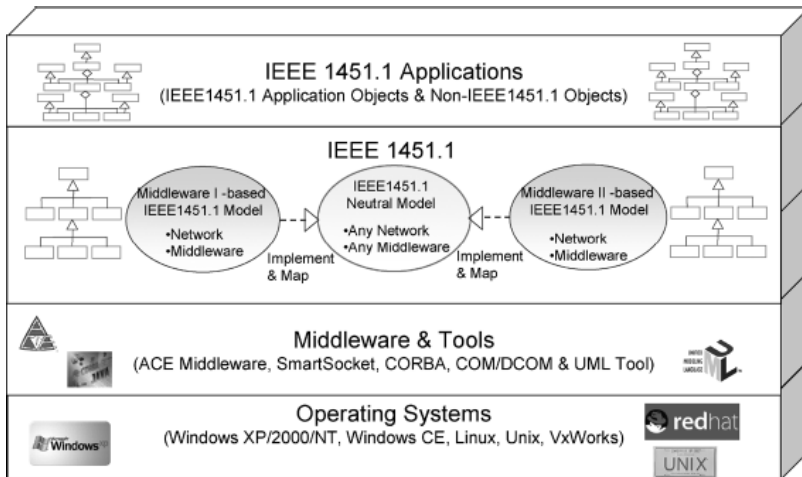


Figura 15: Arquitetura de camadas do framework orientado a objetos para o padrão IEEE 1451.1 (LEE; SONG, 2005).

tado pelo NCAP, e pode ser utilizada por aplicações de medição e controle a fim de acessar a camada IEEE 1451.0. Esta API contém as operações para leitura e escrita dos TransducerChannels e TEDS, assim como operações para o envio de comandos de configuração, controle e execução aos TIMs. A interface de serviço do transdutor na camada IEEE 1451.0 contém cinco interfaces: *TimDiscovery*, *TransducerAccess*, *TransducerManager*, *TEDSManager* e *AppCallback*. As quatro primeiras são implementadas pela camada IEEE 1451.0 e são chamadas pelas aplicações de medição. Caso a aplicação deseje recursos opcionais avançados, será necessário implementar a interface *AppCallback*, a qual será invocada pela camada IEEE 1451.0. A figura 16 apresenta o acesso, através do protocolo HTTP, a um NCAP, o qual se comunica com os TIMs. O símbolo “S” presente no TIM representa um sensor e “A”, um atuador.

Esta dissertação visa utilizar o protocolo SIP para a comunicação entre a rede do usuário e o NCAP. Esta nova abordagem traz benefícios em frente às propostas presentes na norma IEEE 1451 principalmente quando utilizado sensores multimídia, pois IEEE 1451.1 e IEEE 1451.0 HTTP não tratam este tipo de dispositivo, enquanto que SIP representa um protocolo próprio para o estabelecimento de sessões para comunicação interativa entre usuários. Este protocolo é muitas vezes utilizado juntamente com o protocolo RTP, o qual é empregado em aplicações de tempo real, por exemplo, entrega de dados de áudio, como voz sobre IP (do inglês *Voice over Internet Protocol*, VoIP).

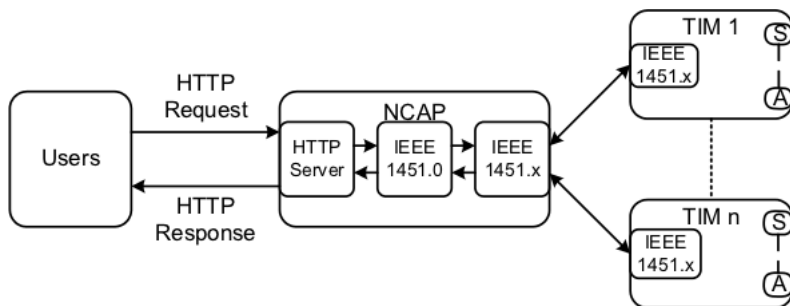


Figura 16: Acesso HTTP a um NCAP (IEEE 1451.0, 2007).

Assim, através de SIP phones e independentemente da localização física dos mesmos, usuários podem coletar informações de sensores, assim como modificar seus dados. A seguir, serão exibidas algumas outras abordagens externas à norma existentes.

Song e Lee apresentam uma nova abordagem para acessar transdutores através da rede a qual chamaram de *Smart Transducer Web Services* (STWS). Este sistema é baseado no padrão IEEE 1451 e consiste em um consumidor de serviços, um provedor de serviços e um nó sensor sem fio. O consumidor e o provedor podem se comunicar entre si utilizando o STWS através de mensagens SOAP (*Simple Object Application Protocol*). Assim, os consumidores neste sistema podem detectar transdutores ativos, acessar dados e recuperar os TEDS dos mesmos. O provedor e consumidor de serviços são gerados a partir de um arquivo WSDL (*Web Service Description Language*), o qual descreve as mensagens, operações e outras informações do serviço Web. Tanto o provedor quanto o sensor sem fio são implementados com base na norma IEEE 1451.0 e IEEE 1451.5, utilizando o protocolo de comunicação sem fio IEEE 802.11 (SONG; LEE, 2007a) (SONG; LEE, 2007b).

Uma continuação da proposta anterior descreve o chamado *Sensor Alert Web Service* (SAWS) para redes de sensores baseadas em IEEE 1451 (LEE; SONG, 2009). O SAWS consiste de um *Sensor Alert Notification Producer Service* (SANPS), representando um serviço de assinatura, e um *Sensor Alert Notification Consumer Service* (SANCS), o qual representa um serviço de notificação, ambos descritos através da linguagem WSDL e baseados no sistema de notificação de serviços Web (do inglês *Web Services Notification*, WS-Notification) (OASIS Standard, 2006a) (OASIS Standard, 2006b) (OASIS Standard, 2006c). Um protótipo do sistema SAWS consistindo de um consumidor, um provedor e um nó sensor sem fio IEEE 1451.5-802.11 é apresentado por Lee. Primeiramente, o consumidor deve se registrar em

um provedor, informando os tópicos que o mesmo deseja monitorar em determinado sensor. Quando uma leitura de dados do sensor excede os limites pré-configurados, o provedor enviará automaticamente ao consumidor uma mensagem de notificação descrita através do protocolo CAP (*Common Alert Protocol*), o qual representa um formato de dados baseado em XML para a troca de avisos entre tecnologias de alerta.

Chunshan propõe o uso de certas mensagens SIP para a comunicação entre sensores conectados a uma rede. Este trabalho utiliza somente os conceitos do padrão IEEE 1451, tais como NCAP, TIM e TEDS, porém, os comandos presentes na norma não são utilizados, sendo os mesmos substituídos por mensagens SIP. A requisição SIP *REGISTER* é utilizada quando um nó deseja entrar na rede, *INFO* inicializa uma sessão, *BYE* permite que um sensor deixe a sessão e as mensagens *UPDATE* e *INFO* modificam os dados dos TEDS. Alguns dos benefícios da utilização do protocolo SIP são: *plug and play* pode ser realizado facilmente através dos princípios de registro SIP, redes híbridas podem ser conectadas através do protocolo e, por fim, usuários podem gerenciar sensores através da Internet, utilizando dispositivos SIP (CHUNSHAN et al., 2009).

5.1 DISCUSSÃO

Primeiramente, este trabalho adotou a ideia de integrar o protocolo SIP à norma IEEE 1451, proposta por Chunshan (CHUNSHAN et al., 2009). Porém, Chunshan não segue fielmente a norma IEEE 1451, deixando de lado grande parte da padronização, como os comandos enviados entre TIM e NCAP, os quais foram substituídos por mensagens SIP. A presente dissertação age de forma diferente, seguindo a norma por completo, ou seja, utilizando os comandos definidos pelo documento IEEE 1451.0 para a comunicação entre TIM e NCAP, sendo o protocolo SIP utilizado somente entre o usuário final e o NCAP, como demonstrado pela figura 1.

A ideia introduzida pelo *Sensor Alert Web Service* (LEE; SONG, 2009), o qual representa um serviço de notificação, foi aproveitada por este trabalho para permitir que um usuário SIP reconheça o estado de conectividade de determinados sensores, ou seja, se os mesmos estão ou não conectados ao NCAP, além de receber os dados sensorizados quando os *motes* estiverem operando no modo TIM-IM. Esta funcionalidade foi realizada através do esquema de notificação assíncrona de eventos, utilizando os métodos *SUBSCRIBE* e *NOTIFY* do protocolo SIP, conforme descrito na seção 4.3.7.

6 INTEGRAÇÃO DO PROTOCOLO SIP À NORMA IEEE 1451

A fim de alcançar o objetivo final deste trabalho, integração do protocolo SIP à norma IEEE 1451 para redes de sensores sem fio, foram executadas tarefas intermediárias. As mesmas foram divididas em três e representam partes essenciais para a obtenção do resultado final desta dissertação. O primeiro passo refere-se ao protocolo SIP, visando sua utilização em sistemas embarcados com restrição de recursos. Assim, foi implementada uma versão SIP miniatura, porém totalmente compatível com a norma original. O segundo experimento apresenta melhorias na comunicação de sensores sem fio para o padrão IEEE 1451. A norma estabelece que o NCAP deve enviar comandos sempre que o mesmo desejar obter novas leituras sensorizadas pelo *mote*. Porém, ao executar desta maneira, os sensores devem permanecer com seus módulos de comunicação ligados, aguardando por requisições vindas do NCAP e, desta forma, consumindo uma quantidade significativa de energia. Logo, é introduzido um novo esquema, onde o WTIM pode reportar novas informações sempre que houver alguma modificação em seu conjunto de leitura, executando de forma autônoma e mantendo o NCAP informado. Por fim, o último passo trata da inclusão de sensores multimídia ao padrão IEEE 1451, o qual não faz nenhuma referência aos mesmos em seus documentos.

6.1 SIP MINIATURA PARA SISTEMAS EMBARCADOS

No decorrer deste trabalho, foi implementada uma adaptação ao protocolo SIP, a fim de permitir sua utilização em sistemas embarcados com restrição de recursos. Os sistemas destino são aqueles em que não há uma pessoa controlando-os. É, por exemplo, o caso de câmeras de segurança, onde uma sessão pode ser inicializada pelo usuário, fazendo uma chamada via SIP phone para o endereço do aparelho, ou pela própria câmera, quando movimento é detectado. Também, a versão desenvolvida é designado para sistemas confinados, não conectados à Internet. Relacionando com a norma IEEE 1451, esta versão executaria nos TIMs, os quais geralmente representam dispositivos com recursos limitados, e não no NCAP, o qual normalmente não possui este problema.

6.1.1 Implementação

A fim de obter uma versão miniatura do protocolo SIP sem comprometer suas funcionalidades, somente um subconjunto de requisições e campos de cabeçalho foi utilizado. A seguir, serão apresentadas as requisições e os cabeçalhos que foram implementados e é discutido o motivo dos demais não serem necessários.

Entre todas as requisições presentes no protocolo SIP, a versão implementada é restrita a somente três: *INVITE*, *ACK* e *BYE*. Tais mensagens representam os métodos requeridos para que uma sessão seja iniciada e finalizada, conforme demonstrado na figura 17. Outras requisições, tais como as presentes na norma original (ROSENBERG et al., 2002), *CANCEL*, *OPTIONS* e *REGISTER*, e as localizadas em RFCs separados, como *REFER*, *SUBSCRIBE*, *NOTIFY*, *MESSAGE*, *UPDATE*, *INFO* e *PRACK*, não foram implementadas. Estes últimos métodos são extensões ao padrão e, desta forma, são opcionais. Consequentemente, os mesmos não são utilizados na adaptação SIP implementada.

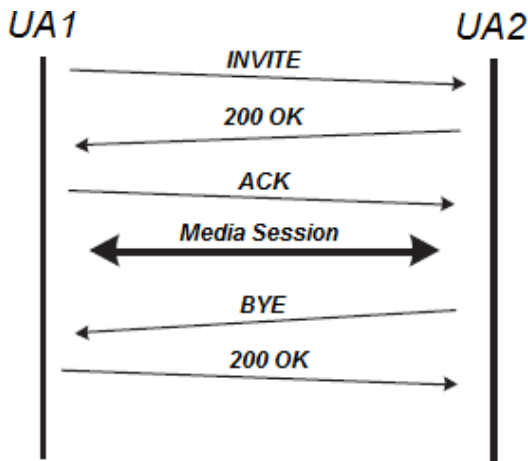


Figura 17: Inicialização e finalização de uma sessão SIP.

Relacionado às requisições SIP do RFC original, *CANCEL* é utilizado para cancelar uma requisição previamente enviada por um UAC. Esta mensagem não é utilizada, pois quando um convite para iniciar a sessão parte do sistema embarcado, não há motivo para cancelar a tentativa de chamada. Já quando a tentativa é feita pelo lado oposto, como o sistema não possui usuário, não há necessidade de enviar respostas provisórias (por exemplo, *180*

Ring) e a conexão deve ser aceita ou rejeitada instantaneamente. Então, o método *CANCEL* pode ser suprimido sem impactar na funcionalidade, pois o mesmo não possui efeito em requisições na qual o UAS já enviou uma resposta final, sendo aplicável em situações onde o lado servidor pode levar algum tempo para responder à requisição.

O método *OPTIONS* é utilizado para questionar um *user agent* ou *proxy server* sobre suas capacidades, sem a necessidade de estabelecer uma chamada com a outra parte. Por exemplo, antes de um UAC enviar uma mensagem *INVITE* com o campo de cabeçalho *Require*, o mesmo pode enviar um método *OPTIONS* para certificar-se de que o UAS suporta o campo requerido. Esta sequência com duas mensagens, *OPTIONS* e *INVITE*, possui o mesmo efeito de enviar *INVITE* duas vezes. Nesta última situação, o primeiro convite requer algumas funcionalidades, entretanto, se a parte destino não suportar a extensão requerida, o procedimento padrão é rejeitar a requisição, enviando um campo de cabeçalho explicando a razão. Desta forma, um segundo *INVITE* pode ser gerado com somente as opções permitidas pelo UAS. Portanto, a requisição *OPTIONS* pode ser removida sem qualquer problema.

REGISTER representa uma mensagem utilizada pelo UA para notificar o servidor SIP de seu URI de contato e um URI o qual deveria ser utilizado para rotear requisições para aquele usuário. Este trabalho suprimiu este método, pois cada UA possui um endereço IP fixo. Desta forma, as mensagens são enviadas diretamente para os IPs dos agentes desejados, sem o uso de servidores (*proxy*, *redirect* e *registrar*).

Dentre os 44 campos de cabeçalho descritos no documento de especificação do protocolo SIP, somente alguns foram incluídos na implementação miniatura apresentada por este trabalho, são eles: *Allow*, *Call-ID*, *Contact*, *Content-Disposition*, *Content-Length*, *Content-Type*, *CSeq*, *From*, *Max-Forwards*, *Record-Route*, *Require*, *Route*, *To*, *Unsupported* e *Via*.

Via, *From*, *To*, *CSeq*, *Call-ID*, *Max-Forwards* e *Contact* são necessários na maioria das mensagens SIP. *Content-Disposition*, *Content-Length* e *Content-Type* identificam o conteúdo presente no corpo da mensagem SIP. *Record-Route* e *Route* provêm informação de roteamento. *Allow*, *Require* e *Unsupported* reportam requisições e campos de cabeçalho suportados ou não pelo *user agent* ou *proxy server*.

Todos os campos de cabeçalho restantes foram excluídos. Um dos principais motivos para a eliminação é que uma parcela deles apenas transmite informações adicionais sobre um usuário a outro. Entretanto, não há necessidade desta troca de dados. Os cabeçalhos que se incluem neste cenário são: *Accept-Language*, *Alert-Info*, *Call-Info*, *Content-Language*, *Date*, *Error-Info*, *Organization*, *Priority*, *Retry-After*, *Server*, *Subject*, *Timestamp*, *User-Agent* e *Warning*.

Accept, *Accept-Encoding*, *Content-Encoding*, *MIME-Version*, *Proxy-Require* e *Supported* foram omitidos da versão miniatura porque não há codificação disponível para eles, ou seja, seus campos ficariam vazios caso fosse necessário enviá-los.

Este estudo não suporta esquemas de autenticação, pois o mesmo é designado para sistemas confinados. Logo, alguns cabeçalhos foram excluídos, como: *Authentication-Info*, *Authorization*, *Proxy-Authenticate*, *Proxy-Authorization* e *WWW-Authenticate*. Os campos de cabeçalho restantes foram removidos por motivos próprios, por exemplo, *Expires* indica o intervalo de tempo que o conteúdo de uma mensagem é válido, sendo que após uma resposta este campo não possui mais sentido. Como as respostas são enviadas imediatamente ao receber uma requisição, sem a necessidade de esperar ações de usuários, este cabeçalho não se aplica. *In-Reply-To* enumera os Call-IDs que uma chamada referencia ou retorna. *Min-Expires* transmite o tempo mínimo de expiração suportado por um *registrar server* em resposta a uma requisição *REGISTER*. Como este trabalho não utiliza o método *REGISTER*, o cabeçalho *Min-Expires* também se torna desnecessário. Finalmente, *Reply-To* especifica um URI que deveria ser utilizado em resposta a uma requisição, porém, como estamos enviando mensagens a partir de endereços IP fixos, não há necessidade de um agente possuir mais de um URI identificando-o.

A versão projetada possui, também, as quatro máquinas de estado necessárias para utilização do protocolo SIP, são elas: *INVITE* client transaction, non-*INVITE* client transaction, *INVITE* server transaction e non-*INVITE* server transaction. A versão SIP miniatura foi implementada no sistema operacional EPOS, o qual foi escolhido por possuir um pequeno footprint de memória, uma pilha de comunicação completa, incluindo UDP (usado na camada de transporte do protocolo SIP), e suporte para redes de sensores (FRÖHLICH; WANNER, 2008), sem interação do usuário.

Assim, com todas as mudanças feitas na versão original do protocolo SIP, há uma grande economia de memória, obtendo uma imagem final do sistema com um tamanho reduzido, podendo ser utilizada até mesmo nos menores e mais simples sistemas embarcados presentes atualmente.

6.1.2 Avaliação

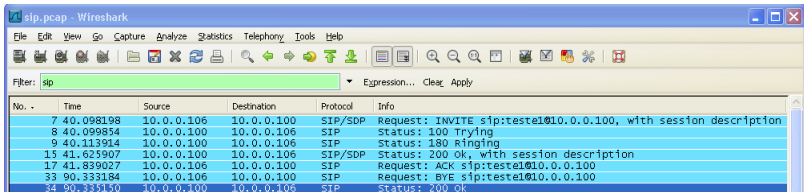
A fim de avaliar a aplicabilidade da implementação SIP para sistemas embarcados realizada, foi medida a biblioteca gerada. A tabela 6 apresenta o tamanho da versão criada, comparando com outras implementações conhecidas, como oSIP library (MOIZARD, 2002) e a parte referente ao SIP do As-

terisk (DIGIUM, 1999). Todos os códigos foram compilados para a arquitetura IA32, utilizando o compilador GNU gcc, versão 4.0.2, e medidos através da ferramenta ia32-size, versão 2.16.1. Os resultados mostram que, com todas as reduções feitas, foi obtida uma biblioteca final com 87.450 bytes, quase três vezes menor do que oSIP library e oito vezes menor do que o Asterisk. O tamanho da imagem final, incluindo EPOS e a versão SIP, alcançou 118.532 bytes (116 Kbytes), muito menos do que os outros, mesmo sem considerar os muitos megabytes de overhead do sistema operacional.

Tabela 6: Tamanho de implementações SIP (em bytes).

Seção	SIP Miniatura	libosip	Asterisk
.text	81.074	232.914	609.678
.data	6.360	3.220	7.912
.bss	16	1.172	67.972
TOTAL	87.450	237.306	685.562

Para demonstrar a funcionalidade, um teste de comunicação entre duas máquinas foi realizado. Em um lado, o sistema operacional EPOS com a versão SIP implementada, rodando em uma máquina virtual VirtualBox. Do outro lado, um computador com um SIP phone pronto para fazer e receber chamadas. Foi testada a inicialização e finalização de sessões entre as duas máquinas. A figura 18 apresenta a saída do teste no formato de fluxo de mensagens, através da interface gráfica do programa Wireshark. O IP 10.0.0.106 representa a máquina virtual com EPOS, enquanto que o endereço 10.0.0.100 roda o sistema operacional Windows XP e o SIP phone X-Lite.



No.	Time	Source	Destination	Protocol	Info
7	40.098198	10.0.0.106	10.0.0.100	SIP/SDP	Request: INVITE sip:testel010.0.0.100, with session description
8	40.099854	10.0.0.100	10.0.0.106	SIP	Status: 180 Trying
9	40.113914	10.0.0.100	10.0.0.106	SIP	Status: 180 Ringing
13	41.825907	10.0.0.100	10.0.0.106	SIP/SDP	Status: 200 OK, with session description
17	41.839027	10.0.0.106	10.0.0.100	SIP	Request: ACK sip:testel010.0.0.100
33	90.333184	10.0.0.106	10.0.0.100	SIP	Request: BYE sip:testel010.0.0.100
34	90.335140	10.0.0.100	10.0.0.106	SIP	Status: 200 OK

Figura 18: Inicialização e finalização de uma sessão SIP no Wireshark.

6.2 MELHORIAS NA COMUNICAÇÃO SEM FIO DO IEEE 1451

Esforços passados para trazer IEEE 1451 para o domínio de redes de sensores sem fio focaram principalmente em interoperabilidade, sem maiores preocupações relacionadas à eficiência e conformidade com as semânticas tradicionais de tais redes (OOSTDYK; MATA; PEROTTI, 2006) (SONG; LEE, 2008). Enquanto transdutores comuns implementados em redes de controle normalmente operam em períodos fixos, ou mesmo como escravos de controladores primários, sensores sem fio evitam tais regimes devido à sua inerente energia e sua ineficiente utilização da largura de banda. A estratégia de reconciliação proposta por Torri (TORRI, 2008) e aprimorada no decorrer do presente trabalho é baseada no chamado *Transducer Interface Module Initiated Message* (TIM-IM). Assim, ao invés de o NCAP fazer polling por novos dados, conforme figura 19, os WTIMs podem reportar novas informações sempre que houver novas leituras sensoriadas, como demonstrado na figura 20, deixando a comunicação entre WTIM e NCAP mais eficiente.

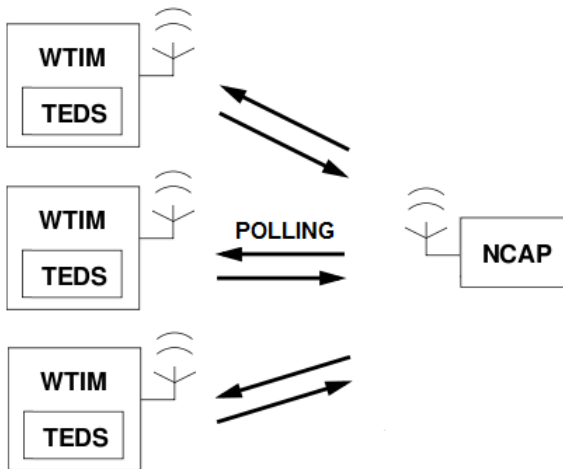


Figura 19: Visão geral da estratégia tradicional (polling) do padrão IEEE 1451.

Neste contexto, WTIMs podem decidir se uma variação observada nos dados sensoriados deve ser reportada para o NCAP via mensagem TIM-IM ou simplesmente ignorada. No pior caso, ou seja, um ambiente modificando constantemente, esta estratégia dispensa as mensagens de requisição enviadas pelo NCAP, desta forma, economizando a banda de rede disponível e também

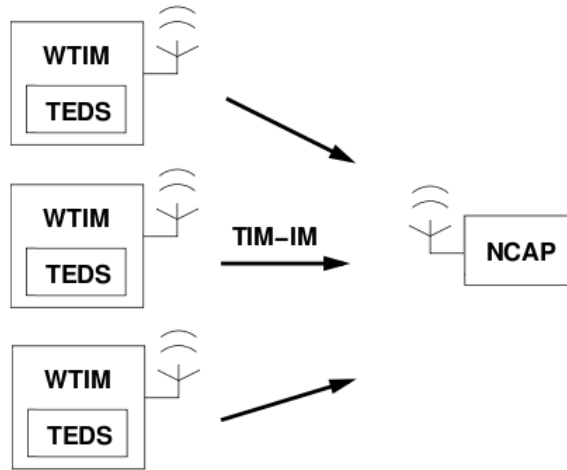


Figura 20: Visão geral da estratégia TIM-IM para RSSF.

a energia do *mote*, pois o mesmo não necessita ouvir constantemente o canal por requisições. No caso de ambientes relativamente estáveis, TIM-IM pode ser largamente eficiente. Esta abordagem é apresentada a seguir.

6.2.1 Implementação

Devido ao problema referente à quantidade de energia disponível em sensores sem fio, algumas aplicações optam por manter os *motes* adormecidos durante a maior parte de sua operação, reconectando-os de tempos em tempos e realizando suas leituras, processamentos e comunicações com outros nós, enviando os dados coletados pelos mesmos. Entre os modos de operação disponíveis na norma IEEE 1451.0, o qual melhor se encaixa neste cenário é chamado *streaming operation*, como discutido na seção 3.5.

Adicionalmente, o padrão IEEE 1451.0 descreve uma estrutura chamada *TIM Initiated Message*. Entretanto, somente demonstra seu uso para enviar informações de estado a partir do TIM para o NCAP, tais como para indicar que algum TEDS foi modificado ou se um comando foi rejeitado. Assim, a norma não descreve como proceder para enviar dados coletados pelo sensor, ou seja, não fornece um modo de o TIM tomar a iniciativa de enviar informações sensorizadas pelo seu TransducerChannel para o NCAP no qual está registrado.

Para isto, este trabalho adotou a estrutura *TIM Initiated Message*, pre-

enchendo os campos existentes com os valores como instruído pelo comando *Read TransducerChannel data-set segment*, descrito pela norma. Desta forma, a aplicação deve esperar por amostras obtidas pelo TransducerChannel, o qual está operando no modo *streaming operation*.

A estrutura da mensagem é apresentada na figura 21. Os dois primeiros octetos são formados pelo endereço do TransducerChannel o qual está enviando a mensagem. Os próximos dois octetos especificam o tipo de comando da mensagem, onde o byte mais significativo indica a classe do comando a ser utilizada e o menos significativo representa a função do mesmo. Este trabalho preenche estes campos com os valores 3 e 1, indicando a classe *transducer operating state* e a função *Read TransducerChannel data-set segment*, respectivamente. A terceira parte também consiste de dois bytes e indica o número de octetos transmitidos a seguir, em outras palavras, quantos octetos são esperados no campo de dados. Finalmente, há os bytes contendo o conjunto de leituras obtidas pelo sensor.

Channel (2 bytes)	Command (2 bytes)	Length (2 bytes)	Data (X bytes)
------------------------------------	------------------------------------	-----------------------------------	---------------------------------

Figura 21: Formato da mensagem TIM-IM.

Os vários TEDS contêm informações relacionadas ao TIM e, sendo assim, representam o melhor lugar para descrever se um sensor está operando no modo polling ou TIM-IM. Um importante TEDS é o *TransducerChannel TEDS*, o qual contém diversas informações, tais como a unidade física utilizada para definir a informação sendo medida (por exemplo, metro, quilograma, segundos, kelvin, etc.), o número de amostras de dados transmitidas pelo sensor, os modos de amostragem e transmissão suportados pelo TransducerChannel, e assim por diante.

A fim de permitir o uso de mensagens iniciadas pelo TIM, foi estendida a tabela 58 do documento IEEE 1451.0 (IEEE 1451.0, 2007), a qual descreve os modos de transmissão suportados pelo TransducerChannel. Esta informação está contida no *TransducerChannel TEDS* e sua nova tabela, com três novas entradas (valores 5, 6 e 7), é apresentada na tabela 7, a qual representa uma cópia da presente na norma, exceto pelas adições aqui realizadas. De acordo com as alterações, um sensor pode descrever que seu TransducerChannel opera exclusivamente no modo TIM-IM, permitindo que o sensor economize sua energia, mantendo-se em um estado de dormência a maior parte do seu tempo. Sucintamente, o valor 1 representa um TransducerChannel executando através do modo polling, valores 2, 3 e 4, rodando nos modos

polling e TIM-IM, e valores 5, 6 e 7, somente TIM-IM.

Tabela 7: Atributos de transmissão de dados.

Valor	Descrição
0	Reservado.
1	Este TransducerChannel é somente capaz de ser operado no modo <i>only when commanded</i> .
2	Este TransducerChannel é capaz de ser operado nos modos <i>Streaming when a buffer is full</i> ou <i>only when commanded</i> .
3	Este TransducerChannel é capaz de ser operado nos modos <i>Streaming at a fixed interval</i> ou <i>only when commanded</i> .
4	Este TransducerChannel é capaz de ser operado nos modos <i>only when commanded</i> , <i>Streaming when a buffer is full</i> ou <i>Streaming at a fixed interval</i> .
5	Este TransducerChannel é capaz de ser operado no modo <i>Streaming when a buffer is full</i> .
6	Este TransducerChannel é capaz de ser operado no modo <i>Streaming at a fixed interval</i> .
7	Este TransducerChannel é capaz de ser operado nos modos <i>Streaming when a buffer is full</i> ou <i>Streaming at a fixed interval</i> .
8-255	Reservado.

A figura 22 apresenta o diagrama de sequência contendo as mensagens trocadas entre NCAP e TIM, operando no modo polling. A figura 23 mostra o diagrama quando utilizado o método de mensagens iniciadas pelo TIM. Resumidamente, uma vez que o WTIM tenha se registrado ao NCAP e este último tenha requisitado os TEDS do WTIM conforme desejado, os seguintes passos, para o modo TIM-IM, são:

1. O NCAP envia um comando do tipo *operate*, a fim de colocar o TransducerChannel do WTIM em operação;
2. O TransducerChannel entra em modo de operação, desconectando-se do NCAP, caso possível;
3. O TransducerChannel começa a fazer as leituras através do sensor, em intervalos regulares, até que o conjunto de dados esteja completo, colocando o WTIM em modo de dormência entre estes intervalos;
4. O WTIM conecta-se novamente, aguarda um comando *operate*, e o conjunto de dados é enviado ao NCAP, utilizando a estrutura de uma

mensagem iniciada pelo TIM e os campos preenchidos seguindo o formato de um comando de leitura, como descrito anteriormente;

5. O WTIM desconecta-se novamente, retomando as operações a partir do terceiro passo.

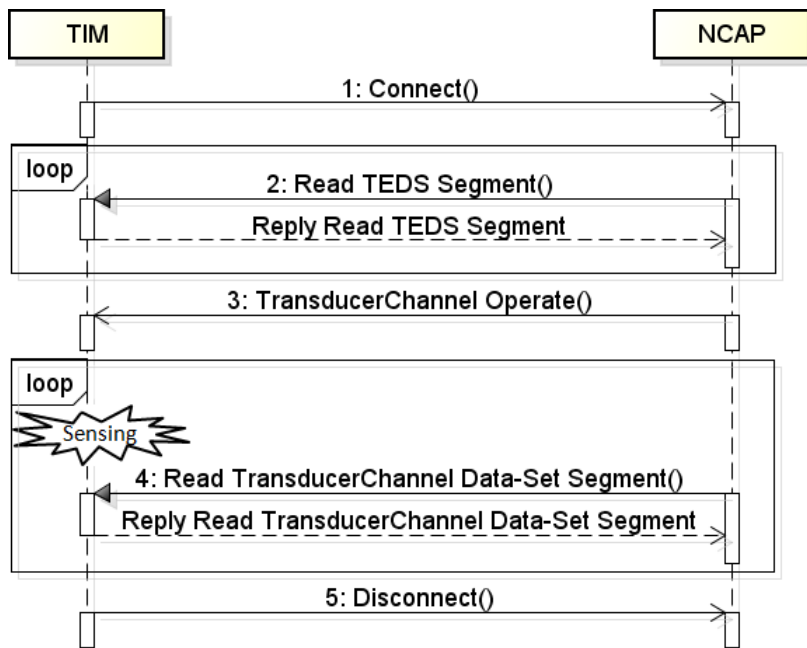


Figura 22: Diagrama de sequência para o modo polling.

De acordo com a figura 23, algumas trocas de mensagens entre NCAP e TIM podem ser removidas a fim de otimizar o sistema, reduzindo o consumo da banda de rede e a energia gasta pelo *mote*. Assim, uma abordagem TIM-IM otimizada é apresentada na figura 24. Sabendo que o TIM opera no modo TIM-IM, o NCAP não enviará nenhuma requisição de dados. Então, as mensagens de conexão, operação e desconexão foram removidas, embora o *mote* estará no estado de dormência durante o tempo de sensoriamento e não estará apto a responder mensagens vindas do NCAP. Esta abordagem otimizada pode ser utilizada em situações onde é útil economizar a banda de rede, porém, NCAP e TIM devem concordar previamente em usar este modo. Logicamente, o NCAP não poderá utilizar mecanismos de *keepalive* (mensagens enviadas a fim de verificar se o enlace de comunicação ainda está

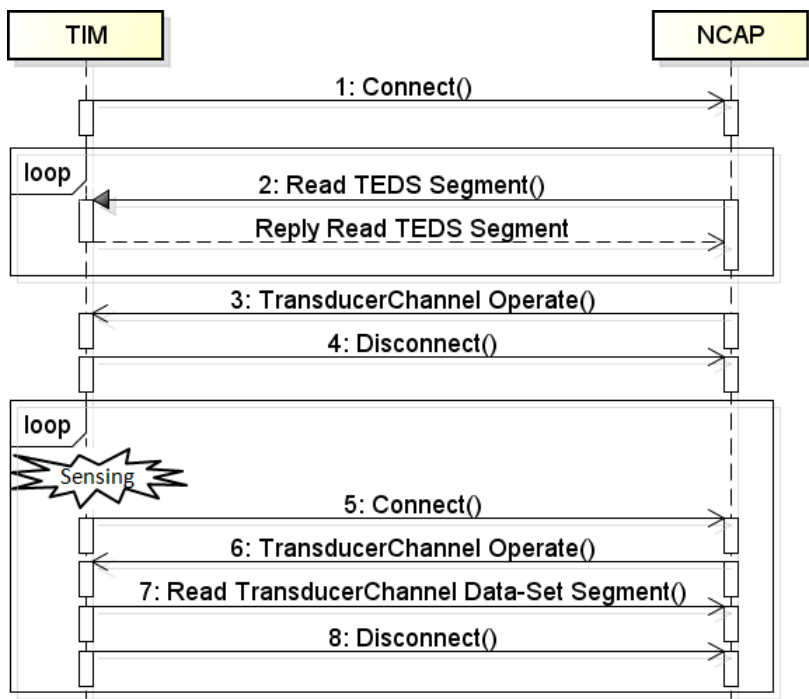


Figura 23: Diagrama de sequência para o modo TIM-IM.

operando) quando os TIMs operam neste modo, pois estas mensagens não serão respondidas, acarretando no término da conexão.

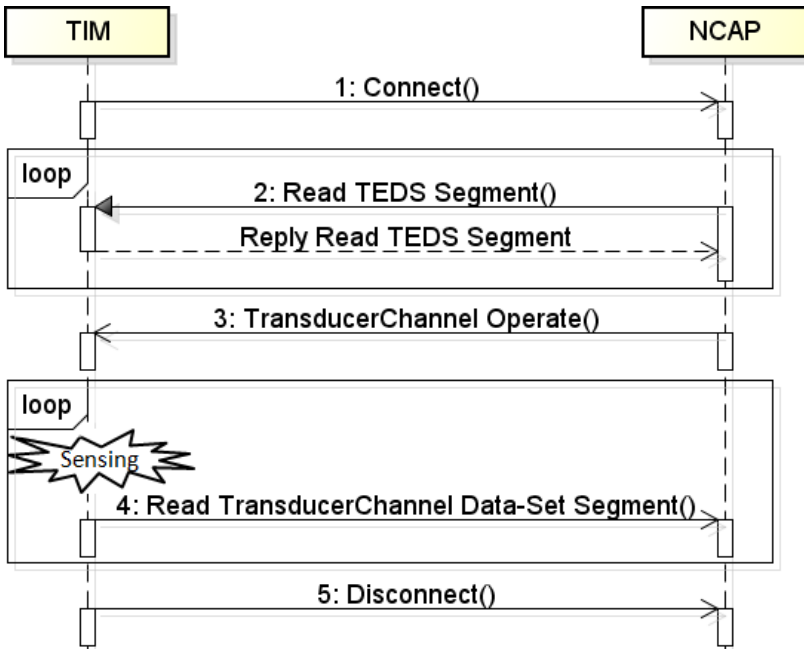


Figura 24: Diagrama de sequência para o modo TIM-IM otimizado.

6.2.2 Avaliação

A fim de validar a estratégia apresentada, foi implementado o padrão IEEE 1451 e executado na plataforma EPOSMoteII, demonstrada pela figura 25. Foi utilizado o sensor de temperatura presente no *mote*, o qual realiza leituras a partir de -40°C até 125°C . Esta plataforma foi escolhida pois a mesma foca em modularização e possui módulos intercambiáveis, facilitando a inclusão do padrão IEEE 1451.

Foi utilizado como plataforma de software o sistema EPOS. O mesmo foi escolhido por possuir um pequeno footprint de memória, uma pilha de comunicação completa e suporte à rede de sensores. Ainda, EPOS implementa os protocolos TCP/IP, os quais foram utilizados para comunicação entre NCAP e TIM, desta forma, garantindo a entrega dos pacotes. Por último,

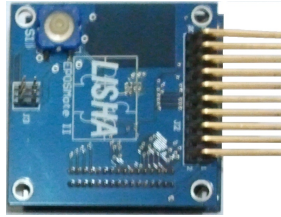


Figura 25: Plataforma EPOSMoteII.

C-MAC evita a escuta ociosa (*idle listening*) implementando a escuta em baixa potência (*low power listening*), onde o mesmo, periodicamente, ativa o rádio para checar por atividades no canal e, se alguma ação é detectada, o rádio é mantido ligado a fim de receber um pacote, de outra forma, ele é imediatamente desligado por algum tempo (STEINER; MÜCK; FRÖHLICH, 2010).

A figura 26 apresenta a estrutura TIM-IM preenchida. Como este teste utiliza somente um TransducerChannel, o valor do primeiro campo é igual a 1. Em seguida, há dois octetos especificando o tipo de comando da mensagem, preenchidos com os valores 3 e 1, indicando a classe *transducer operating state* e a função *Read TransducerChannel data-set segment*. Os conjuntos de dados são compostos de 10 leituras de temperatura, cada uma contendo um número em ponto flutuante de 32 bits (4 bytes). Desta forma, para o campo indicando o tamanho é associado o valor 40 (0x28 em hexadecimal). Finalmente, há os 40 octetos contendo a sequência de leituras obtidas pelo sensor de temperatura.

O propósito desta avaliação é determinar a sobrecarga do modo de operação tradicional (polling) comparado ao TIM-IM, de acordo com a norma. O cenário de teste é apresentado na figura 27 e os resultados da avaliação de energia são mostrados na figura 28.

Durante os primeiros segundos, os três casos possuem o mesmo comportamento, pois este é o tempo para que o TIM se auto-identifique e responda às requisições de TEDS iniciadas pelo NCAP. Depois disto, o método polling, o qual implementa *low power listening* com um timeout de 500 ms e um período de dormiência de 1000 ms (valores default do EPOS), continua a consumir energia, pois o mesmo mantém o rádio ativo durante parte do seu tempo, esperando por mensagens. Em ambos os métodos TIM-IM, o sensor pode desligar o rádio enquanto coleta seus dados, religando de tempo em tempo para enviar a informação. O modo TIM-IM otimizado economiza um pouco mais de energia, pois o mesmo não necessita enviar os pacotes TCP responsáveis pela conexão e desconexão do sensor cada vez que o mesmo for

TransducerChannel Number	msb	0x00
	lsb	0x01
Command	msb	0x03
	lsb	0x01
Length	msb	0x00
	lsb	0x28
Data	1	0x??
	...	0x??
	40	0x??

Figura 26: Formato da mensagem TIM-IM preenchido.

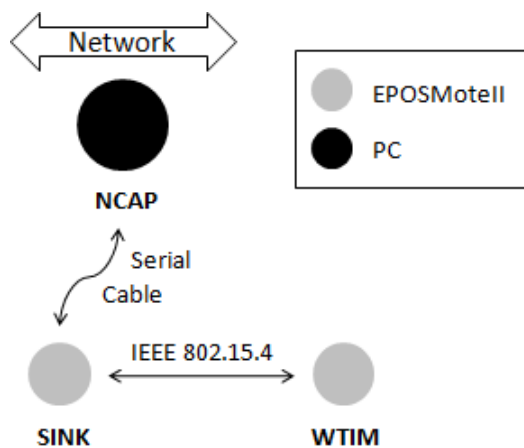


Figura 27: Cenário de teste para o modo TIM-IM

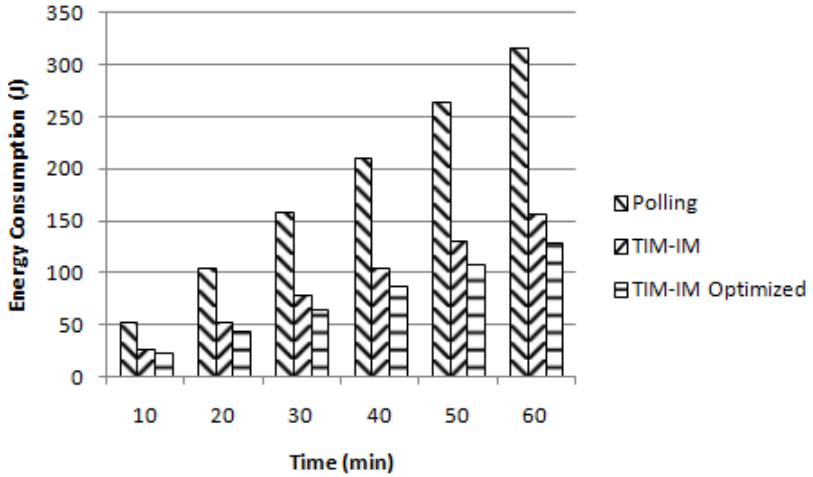


Figura 28: Consumo de energia para o sensor de temperatura.

enviar os dados sensorizados.

A fim de não favorecer nenhum dos três casos, o TransducerChannel coleta uma amostra de temperatura a cada 10 segundos em todos os cenários e, nos modos TIM-IM, o *mote* envia as leituras autonomamente quando seu buffer com dez entradas está cheio, ou seja, a cada 100 segundos. O método polling funciona de forma similar, porém, o NCAP é responsável por enviar uma requisição a cada 100 segundos ao TIM. Os testes foram realizados possuindo a mesma distância entre NCAP e TIM. A corrente de operação teórica do EPOSMoteII é apresentada abaixo:

- 29 mA: rádio no modo TX
- 22 mA: rádio no modo RX
- 3.3 mA: rádio desligado, CPU ativa
- 0.85 mA: rádio desligado, CPU inativa

Os resultados mostram que depois de 1 hora rodando, o método TIM-IM consumiu menos do que a metade da quantidade de energia do sensor comparado ao modo polling (129,83 J no TIM-IM otimizado, 156,13 J no TIM-IM e 316,96 J no polling). Isto representa uma significativa redução no consumo de energia usando TIM-IM, consequentemente aumentando a vida útil da bateria do sensor. Utilizando técnicas mais sofisticadas, tal como enviar os dados somente quando houver alguma mudança nas amostras de

temperatura coletadas comparada ao conjunto de dados anterior (modificação facilmente feita em software), em situações onde não há muita variação nos dados sensorizados, um ganho ainda maior pode ser alcançado.

Outro problema relacionado a sensores sem fio é que os mesmos são, normalmente, sistemas embarcados com restrição de recursos. Assim, os programas desenvolvidos devem ser pequenos o suficiente para caber neles. A fim de analisar o tamanho de memória da implementação realizada, foi utilizado a ferramenta GNU arm-size, versão 2.20. Os resultados são mostrados na tabela 8. A primeira linha, *WTIM (Temperatura) + EPOS*, mostra o tamanho do sistema operacional EPOS juntamente com a implementação do WTIM que executa o sensoriamento de temperatura. A última linha, *WTIM (Temperatura)*, apresenta o tamanho da implementação do WTIM somente, sem o EPOS.

Tabela 8: Tamanho da implementação do WTIM (em bytes).

Seção	.text	.data	.bss	TOTAL
WTIM (Temperatura) + EPOS	56.492	213	5.280	61.985
WTIM (Temperatura)	8.320	0	4	8.324

6.3 SUPORTE A REDES DE SENSORES MULTIMÍDIA SEM FIO NO IEEE 1451

Embora WMSN possua um potencial para habilitar muitas aplicações, por exemplo, na área de segurança, sistemas de controle de tráfego, prestação de cuidados médicos avançados, dispositivo de estacionamento automatizado, casas inteligentes, monitoração de ambiente e assim por diante (AKYILDIZ; MELODIA; CHOWDURY, 2007), o padrão IEEE 1451 não lida com este tipo de sensores. Então, este trabalho apresenta uma adaptação à norma IEEE 1451, a fim de permitir sua utilização em redes de sensores multimídia sem fio. Isto foi alcançado através da inclusão de algumas informações nos TEDS, tal como a descrição do fluxo de dados (áudio ou vídeo) e, também, adicionando dois novos comandos ao conjunto de mensagens descrito pela norma, objetivando notificar o lado receptor de que um fluxo multimídia está iniciando/terminando. A seguir é explicado em detalhes a abordagem implementada.

6.3.1 Implementação

Devido ao crescente uso de sensores multimídia e sua potencial utilização em muitas áreas, este trabalho apresenta o suporte a WMSN ao padrão IEEE 1451. Como já comentado, *TransducerChannel TEDS* contém diversas informações relacionadas ao sensor, tal como a unidade física utilizada para definir a informação sendo medida. A fim de permitir o envio de informações multimídia, a tabela 48 do documento IEEE 1451.0 (IEEE 1451.0, 2007), a qual descreve a estrutura do bloco de dados do *TransducerChannel TEDS*, foi estendida. A nova tabela não modifica as informações já existentes, somente adicionando duas novas entradas, as quais são mostradas na tabela 9. De acordo com as inclusões, um sensor pode descrever que seu *TransducerChannel* envia dados multimídia (fluxo de áudio ou vídeo). Estas novas opções comportam-se como as alternativas não-multimídia existentes, representando duas novas entradas ao campo que descreve as unidades físicas.

Tabela 9: Estrutura do bloco de dados do *TransducerChannel TEDS*

Campo	Nome	Descrição	Tipo	#
61	AudioStream	O expoente para Audio Stream	UInt8	1
62	VideoStream	O expoente para Video Stream	UInt8	1

Há casos onde não é possível expressar completamente a unidade física com somente um campo. Para resolver este problema, é fornecido pela norma um TEDS opcional baseado em texto, chamado *Units Extension TEDS*, possibilitando a inclusão de um texto visando estender o campo que descreve as unidades (IEEE 1451.0, 2007). O *TransducerChannel TEDS* possui um espaço onde é possível fornecer o código de acesso deste TEDS de extensão. Este trabalho utiliza o *Units Extension TEDS* para reportar algumas informações adicionais sobre o fluxo de áudio ou vídeo sensoriado. Por exemplo, o TEDS baseado em texto pode indicar que o áudio coletado é enviado usando o codec G.711 A-law.

A fim de notificar o lado oposto de que um fluxo de dados está iniciando ou finalizando, a tabela 31 da norma IEEE 1451.0 (IEEE 1451.0, 2007), a qual descreve os comandos utilizados quando o *TransducerChannel* está no estado operacional, foi estendida. A tabela 10 – a qual representa uma cópia da tabela presente no padrão, exceto pelas adições aqui realizadas – apresenta dois novos comandos relativos ao estado operacional do transdutor. As linhas de comandos com números de identificação 5 e 6 foram incluídas neste trabalho e podem ser utilizadas tanto para transdutores operando no

método polling, onde o NCAP requisita os dados sensorizados pelos *motes*, como no modo TIM-IM, onde os sensores são responsáveis por tomar a iniciativa de comunicação com o NCAP ao qual pertencem.

Alguns diagramas de sequência são mostrados abaixo, apresentando a troca de mensagens entre NCAP e TIM, utilizando o padrão IEEE 1451. Em todos os casos o início é o mesmo. Primeiro, o WTIM registra-se ao NCAP e este último requisita os TEDS do *mote* conforme desejado. Então, o NCAP envia um comando do tipo *operate* a fim de colocar o TransducerChannel do TIM em operação. A figura 29 apresenta a abordagem tradicional quando utilizado o modo de operação polling, onde o NCAP deve enviar uma mensagem de requisição para cada resposta do TIM. Como sensores multimídia enviam um fluxo de dados, com possivelmente muitos pacotes, o NCAP também necessita enviar um enorme número de requisições. A fim de otimizar esta situação, a figura 30 apresenta um método polling alternativo, onde o NCAP deve somente enviar um comando para iniciar o fluxo (*Start Read TransducerChannel data-set segment*, tabela 10) e uma outra mensagem a fim de pará-lo (*Stop Read TransducerChannel data-set segment*, tabela 10). Durante este meio tempo, o TIM é responsável por enviar os dados multimídia.

Um sensor operando no modo TIM-IM deve adquirir dados e transmiti-los ao NCAP quando desejado, sem a necessidade de receber comandos adicionais a partir deste. Assim, as leituras feitas pelo sensor podem ser enviadas ao NCAP autônoma e automaticamente. A figura 31 apresenta o diagrama de sequência ao utilizar o modo TIM-IM. Quando o TransducerChannel está ativo e necessita enviar informações multimídia ao NCAP, o sensor inicia enviando um comando para preparar o outro lado (*Start Read TransducerChannel data-set segment*) e então transmite os dados sensorizados, terminando com uma mensagem finalizadora (*Stop Read TransducerChannel data-set segment*). Quando operando no modo TIM-IM, os comandos de inicialização, leitura e término são enviados utilizando a estrutura *TIM Initiated Message*, descrita pela norma IEEE 1451.0.

De acordo com a figura 31, algumas trocas de mensagens entre NCAP e TIM podem ser omitidas a fim de otimizar o sistema, reduzindo o consumo da banda de rede e a bateria do sensor. Assim, é apresentada uma abordagem TIM-IM otimizada, mostrada na figura 32. Como o NCAP sabe que o TIM opera no modo TIM-IM, o mesmo não enviará qualquer requisição de dados. Desta forma, as mensagens indicando conexão, operação e desconexão foram suprimidas, embora o TIM permanecerá no estado de dormência durante o período de sensoriamento, não respondendo a mensagens vindas do NCAP. Esta abordagem otimizada pode ser utilizada em situações onde é útil economizar a banda da rede, porém, NCAP e TIM devem, previamente, concordar em utilizar este modo.

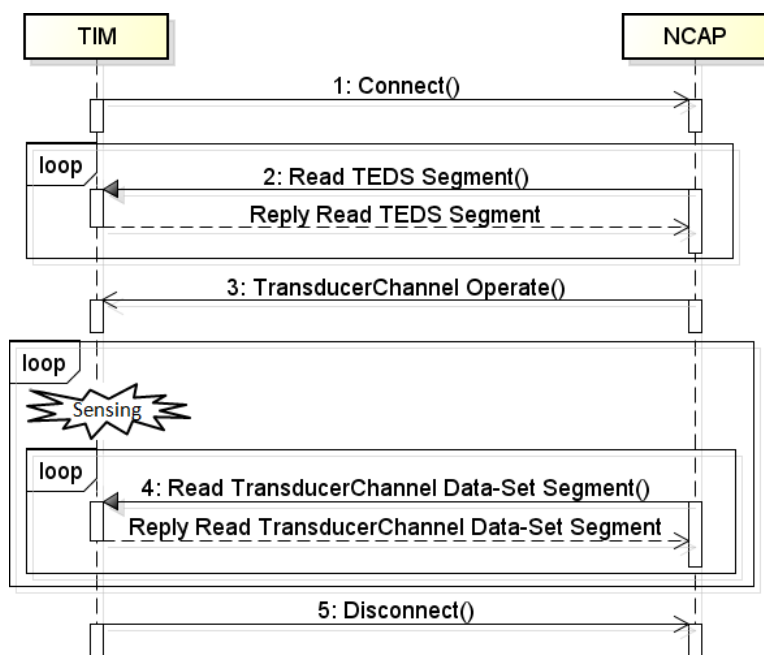


Figura 29: Diagrama de sequência para o modo polling utilizando sensor multimídia.

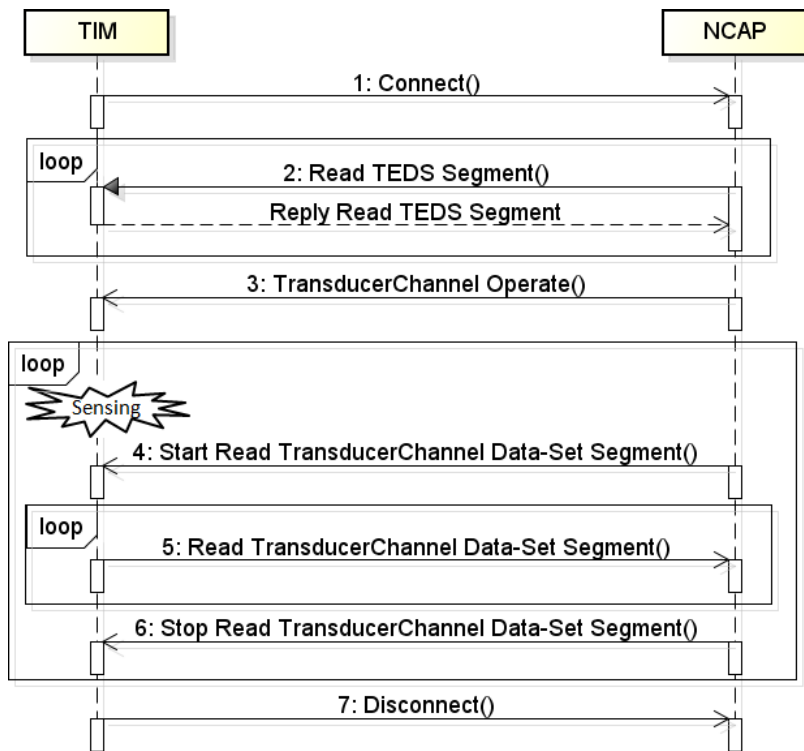


Figura 30: Diagrama de sequência para o modo polling otimizado utilizando sensor multimídia.

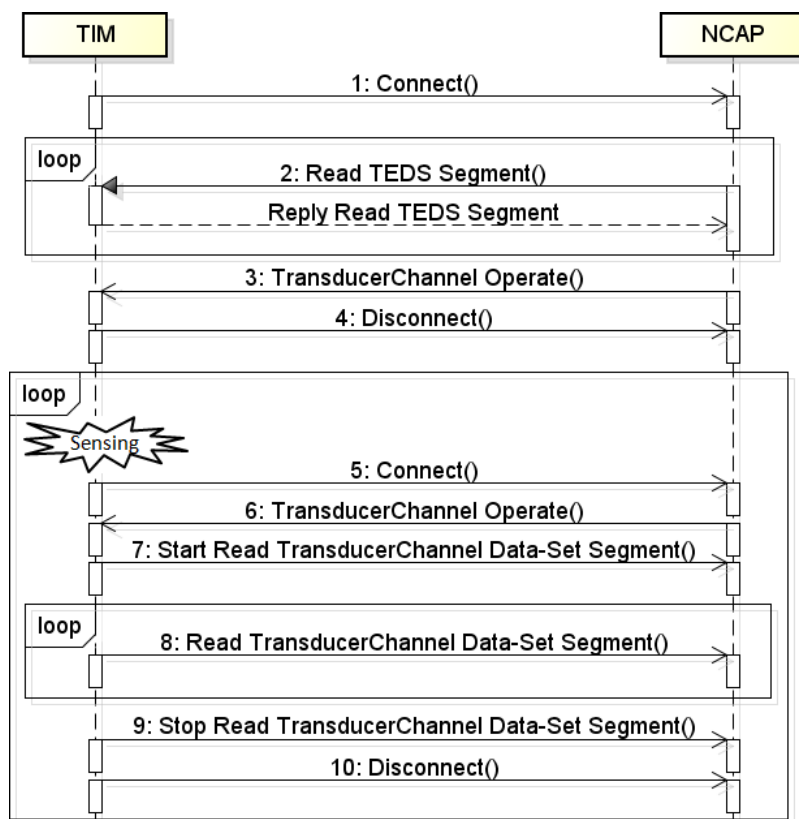


Figura 31: Diagrama de sequência para o modo TIM-IM utilizando sensor multimídia.

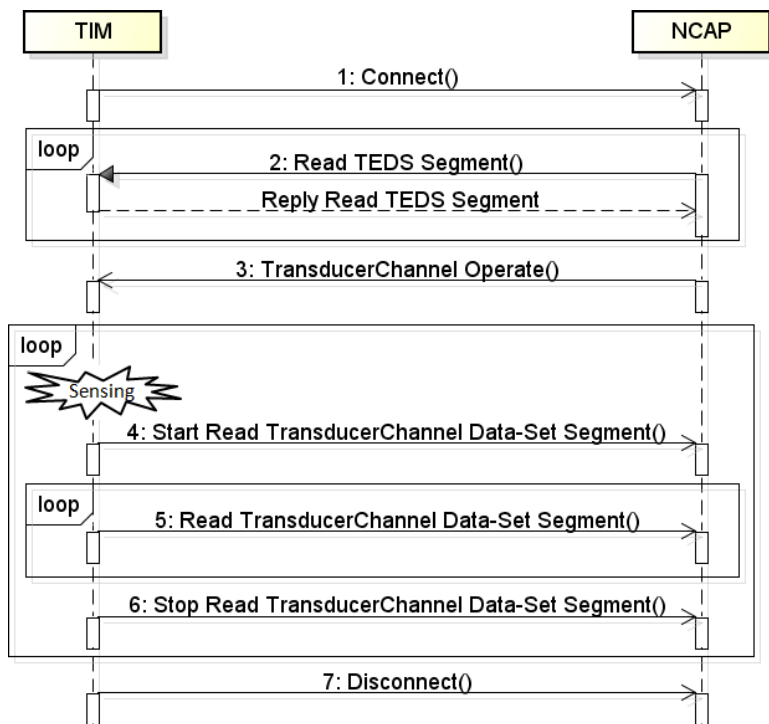


Figura 32: Diagrama de sequência para o modo TIM-IM otimizado utilizando sensor multimídia.

6.3.2 Avaliação

Assim como na seção 6.2, a avaliação da implementação foi testada utilizando o nó sensor EPOSMoteII. A fim de obter um sensor multimídia, foi implantada uma placa de áudio externa, a qual contém um microfone capaz de capturar o áudio circundante, conforme apresentada na figura 33. Foi optado pelo microfone, ao invés de uma câmera de vídeo, devido à sua simplicidade, já que não há nenhuma diferença na utilização destes dois tipos de sensores multimídia perante a norma IEEE 1451. A plataforma de software EPOS também foi adotada, utilizando o protocolo UDP/IP para a transmissão do fluxo de áudio e TCP/IP para garantir a confiabilidade de entrega das outras mensagens.

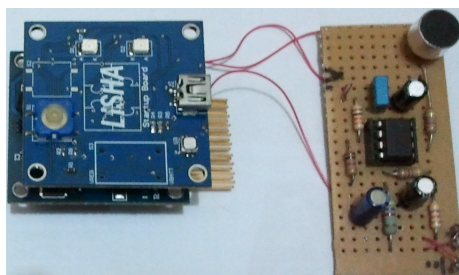


Figura 33: Plataforma EPOSMoteII com microfone.

O propósito desta avaliação é verificar o consumo de energia para sensores multimídia operando nos modos polling e TIM-IM. O cenário de teste é o mesmo apresentado pela figura 27. A figura 34 apresenta o consumo de energia para o sensor multimídia. Neste experimento, o sensor coleta uma amostra de áudio a cada 127 microsegundos, resultando em 7884 amostras por segundo. O nó envia 27 pacotes por segundo, sendo que cada pacote contém 292 amostras. Foi utilizado o codec G.726 com a taxa de bits de 16 kbit/s, o qual representa uma técnica ADPCM (*Adaptive Differential Pulse-Code Modulation*) (International Telegraph And Telephone Consultative Committee, 1990). As amostras são enviadas por 5 segundos, seguido por uma pausa de 5 segundos e retornando ao envio, formando assim um loop. Os quatro modos de operação foram implementados como apresentados neste trabalho.

O consumo é maior aqui quando comparado ao sensor de temperatura devido à diferença na quantidade de dados sendo tratados e também pela necessidade do sensor de áudio realizar um grande número de leituras do ADC,

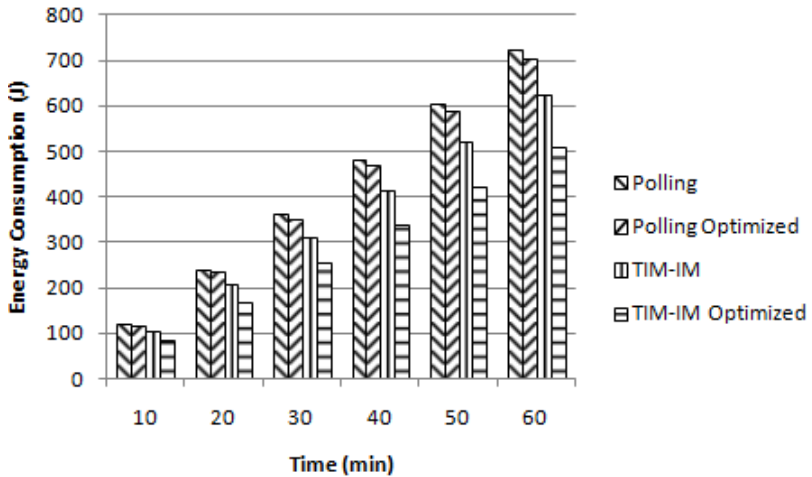


Figura 34: Consumo de energia para o sensor multimídia.

além de executar o codec. A diferença entre os modos polling e TIM-IM é menor para o sensor multimídia, pois o *mote* envia pacotes por 5 segundos com intervalos de 5 segundos, enquanto que o sensor de temperatura envia somente um pacote com intervalos de 100 segundos. Desta forma, o tempo que o rádio pode permanecer desligado é menor para o microfone, o que reduz o espaço para otimização.

Vale ressaltar que ao falar do modo de execução TIM-IM para sensores multimídia, não se trata dos cenários com fluxos de dados incessantes, tal como uma câmera de segurança enviando imagens continuamente. Este tipo de fluxo não necessitaria utilizar TIM-IM, pois uma única sessão poderia ser estabelecida e os pacotes seriam enviados infinitamente. Logo, ao falar em TIM-IM, refere-se àqueles sensores que captam dados, realizam alguma verificação sobre os mesmos e, possivelmente, transmitem-nos. Um exemplo seria uma câmera de segurança a qual só estabeleceria um diálogo com o NCAP caso algum movimento fosse detectado, finalizando a sessão após a normalização das imagens obtidas.

A fim de analisar o footprint de memória da implementação do sensor de áudio, foi utilizado a ferramenta *arm-size*, da GNU Binutils, versão 2.20. A tabela 11 apresenta os resultados. A primeira linha, *WTIM (Áudio) + EPOS*, mostra o tamanho do sistema operacional EPOS, junto com a implementação do WTIM multimídia. A segunda, *WTIM (Áudio)*, apresenta o tamanho do WTIM multimídia, sem o EPOS.

Tabela 11: Tamanho da implementação do WTIM multimídia (em bytes).

Seção	.text	.data	.bss	TOTAL
WTIM (Áudio) + EPOS	62.812	213	5.280	68.305
WTIM (Áudio)	9.332	0	4	9.336

6.4 INTEGRAÇÃO: SIP X IEEE 1451

Após a introdução do modo de execução TIM-IM, permitindo a utilização de sensores sem fio autônomos frente a norma, e com a inclusão de sensores multimídia ao padrão IEEE 1451, além do estudo e implementação do protocolo SIP, esta dissertação visa unir todos estes trabalhos intermediários, apresentando um meio completo para acessar sensores inteligentes, tanto escalares quanto multimídia, através da rede do usuário, utilizando SIP. A seguir, será apresentada a solução desenvolvida. Primeiramente, será discutido como um usuário SIP poderá saber se um sensor está conectado ou não ao NCAP. Em seguida, será apresentado como funciona o acesso aos dados sensorizados por sensores escalares e multimídia, e também o tratamento das operações sobre os TEDS. Para finalizar, os resultados da integração são discutidos.

6.4.1 Notificação do Estado de Conectividade

Como discutido na seção 4.3.7, o protocolo SIP possui uma extensão a qual possibilita seu uso para notificações assíncronas de eventos a usuários remotos. Esta funcionalidade, descrita pelas requisições SIP *SUBSCRIBE* e *NOTIFY*, pode ser utilizada para notificação de presença, permitindo reconhecer o estado de conectividade de um usuário, ou seja, se o mesmo está *on-line* ou *off-line*.

A fim de obter o estado de um TransducerChannel, ou seja, a informação de que o mesmo está conectado ou não ao NCAP, este trabalho utiliza o sistema de notificação de eventos assíncronos do protocolo SIP, através das mensagens *SUBSCRIBE* e *NOTIFY*. Com elas, um usuário de um SIP phone pode requisitar notificações indicando o estado de conectividade de determinados sensores. A figura 35 apresenta o diagrama de sequência demonstrando as mensagens trocadas entre TIM, NCAP e SIP phone para a notificação do estado do *mote*.

Primeiramente, o usuário SIP deve se registrar ao TransducerChannel desejado, enviando uma mensagem *SUBSCRIBE* ao NCAP. Conforme

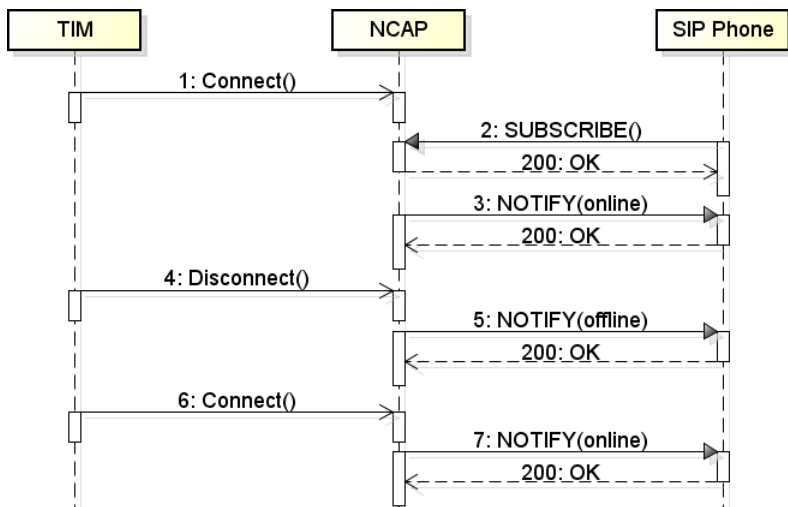


Figura 35: Diagrama de sequência para notificação de estado (*on-line* ou *off-line*).

descrito pela norma referente à notificação de eventos SIP, ao receber uma assinatura, deve-se imediatamente enviar um método *NOTIFY*, informando o estado atual do objeto monitorado, neste caso representado pelo sensor. Posteriormente, ao ocorrer a conexão ou desconexão do *mote*, novas requisições *NOTIFY* são enviadas visando notificar o usuário remoto.

As mensagens originadas pelo usuário SIP com destino ao NCAP devem informar o endereço do sensor ao qual se deseja acessar no campo de destino da requisição SIP. O protocolo especifica o chamado SIP URI, o qual identifica um recurso de comunicação, contendo informação suficiente para iniciar e manter uma sessão com o recurso (ROSENBERG et al., 2002). O URI descrito pelo protocolo SIP possui o seguinte formato¹:

- *sip:user@host*

O campo *user* representa o identificador de um determinado recurso dentro do domínio sendo endereçado. O campo *host* pode conter um nome de domínio totalmente qualificado ou no formato de endereço numérico IPv4 ou IPv6. Ao tentar acessar um sensor específico a partir de um SIP phone, o usuário deve preencher o campo *host* com o endereço do NCAP e a parte *user*

¹Há outras informações presentes no URI do protocolo SIP, porém foram omitidas por não serem necessárias para este trabalho.

com o nome de identificação do transdutor desejado, informação presente no *User's Transducer Name TEDS*, o qual tem como objetivo fornecer um espaço para o usuário do transdutor armazenar o nome pelo qual o sistema reconhecerá o mesmo.

6.4.2 Sensores Não-Multimídia

O protocolo SIP tem como objetivo principal estabelecer sessões entre usuários a fim de trafegar fluxos de dados. Porém, sensores que coletam dados escalares, tais como temperatura, aceleração e etc., não necessitam de uma sessão para enviar seus dados. Pensando nisto, este trabalho utiliza a requisição SIP *MESSAGE*, descrita na seção 4.3.8, permitindo a transferência de mensagens instantâneas entre o usuário SIP e NCAP, sem a necessidade de inicializar uma sessão.

Assim, quando um determinado sensor estiver operando no modo polling, o usuário SIP poderá enviar uma requisição *MESSAGE* ao NCAP requisitando os dados sensorizados pelo *mote*. Ao receber esta mensagem, o NCAP deverá enviar um comando no formato do padrão IEEE 1451 ao Transducer-Channel correspondente, de acordo com o endereço de destino do método SIP. Este comando é chamado de *Read TransducerChannel data-set segment* pela norma. Assim que uma resposta é recebida, o NCAP encaminhará uma nova mensagem SIP do tipo *MESSAGE*, com as informações coletadas pelo sensor, para o usuário SIP que havia requisitado previamente. A figura 36 apresenta o diagrama de sequência contendo as trocas de mensagens descritas.

Quando o TransducerChannel estiver operando no modo TIM-IM, o sensor poderá reportar seus dados de forma autônoma, enviando-os através da mensagem *Read TransducerChannel data-set segment* ao NCAP. Este deverá redirecionar a informação, utilizando a requisição SIP *MESSAGE*, aos usuários previamente registrados àquele *mote* através do esquema de notificação do estado de conectividade, realizado pela mensagem SIP *SUBSCRIBE*. Este modo de funcionamento é demonstrado pelo diagrama de sequência da figura 37.

Os métodos *MESSAGES* transportam conteúdo em seu corpo no formato MIME, tais como *plain/text*, *message/cpim*, *text/html*, entre outros. O presente trabalho não impõe nenhuma restrição quanto ao uso destes modos, possibilitando ao usuário SIP e ao NCAP transmitir o conteúdo no formato de sua escolha, porém, é claro, devendo o mesmo ser suportado pelo lado remoto.

Read TransducerChannel data-set segment representa o nome do comando utilizado para ler um segmento do conjunto de dados do Transduc-

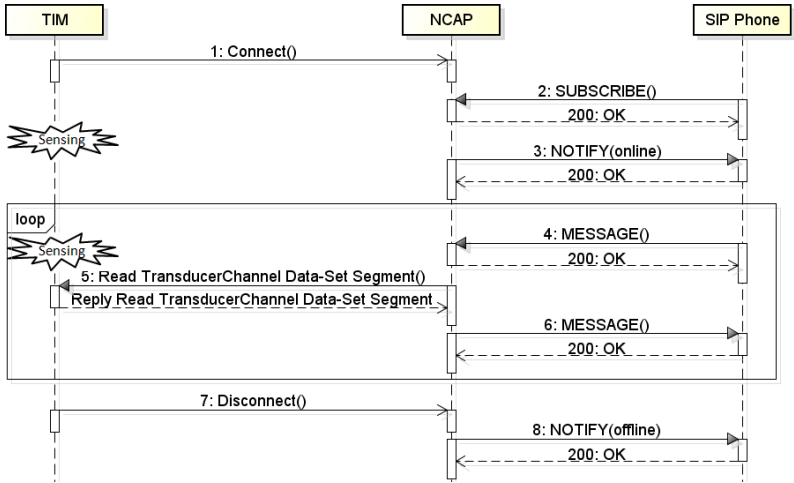


Figura 36: Diagrama de sequência para sensores não-multimídia utilizando o modo polling.

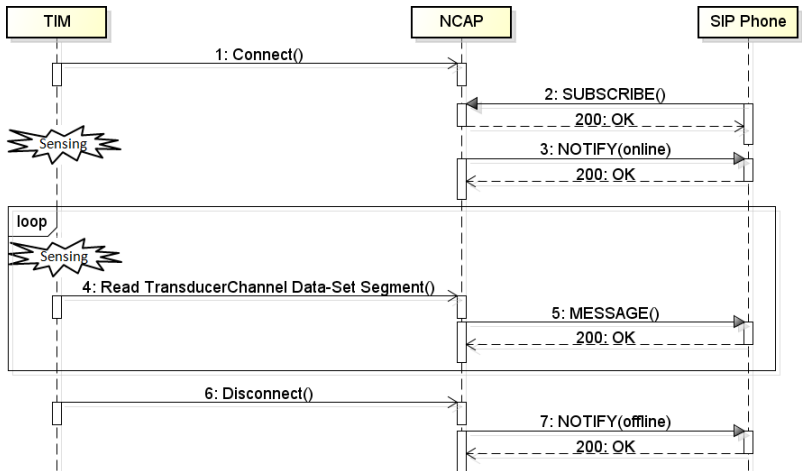


Figura 37: Diagrama de sequência para sensores não-multimídia utilizando o modo TIM-IM.

erChannel no padrão IEEE 1451, podendo ser enviado somente quando o transdutor está no estado operacional. Este comando contém apenas um argumento, o qual especifica o deslocamento do início da leitura dentro do conjunto de dados. Assim, este trabalho utiliza a requisição *MESSAGE*, enviada pelo SIP phone ao NCAP quando utilizado o modo de operação polling, contendo uma frase em seu corpo a fim de identificar que a mensagem enviada corresponde a uma requisição de leitura do conjunto de dados do sensor. Esta frase possui o seguinte formato: *Read TransducerChannel data-set segment <DataSetOffset>*, onde *<DataSetOffset>* deve ser substituído pelo deslocamento desejado referente à leitura que será realizada. Como o método SIP possui como endereço de destino a referência ao TransducerChannel que se deseja acessar, ao receber uma mensagem neste formato, o NCAP possuirá todas as informações necessárias para encaminhar o comando de leitura de dados ao TransducerChannel alvo.

O comando de leitura de dados sensorizados deve gerar uma resposta por parte do sensor contendo dois argumentos: o deslocamento utilizado na leitura dos dados, em muitos casos representando o mesmo valor enviado na requisição, e os blocos de dados lidos do sensor. Assim, ao receber uma mensagem deste tipo, o NCAP deverá encaminhar a mesma ao usuário SIP ao qual executou a requisição, caso seja uma resposta relativa ao modo polling, ou deverá enviar aos SIP phones registrados ao TransducerChannel de origem, caso seja uma mensagem TIM-IM. Isto será realizado através de uma nova mensagem SIP *MESSAGE*, contendo a string *Read TransducerChannel data-set segment Reply <ReadSensorOffset> <ReadSensorData>*, onde os argumentos serão preenchidos com os mesmos dados enviados pelo sensor ao NCAP. Porém, a mensagem SIP utilizada trata as informações em formato texto, enquanto que o padrão IEEE 1451 utiliza os octetos, os quais representam grupos de 8 bits, em formato numérico. Assim, o NCAP é responsável por realizar as transformações necessárias antes de repassar as mensagens tanto para o TIM quanto para o SIP phone. Isto é necessário pois os dados trafegados utilizando a representação numérica feita pelo padrão IEEE 1451 podem conter o valor 0, o qual indica o final de uma string no formato texto, além de poderem conter outros valores que não possuem significado ou que podem causar a apresentação de forma indesejada ao usuário SIP.

6.4.3 Sensores Multimídia

Ao lidar com sensores multimídia, ou seja, os quais trafegam fluxos de dados, como áudio e vídeo, o presente trabalho utiliza o estabelecimento de sessões SIP, o qual consiste em um *three-way handshake* através da troca

de uma requisição *INVITE*, uma resposta contendo a informação 200 OK e um método *ACK* finalizando o processo, conforme descrito na seção 4.3.1. O fluxo de dados é enviado por pacotes RTP, o qual é bastante utilizado junto ao protocolo SIP.

Para o envio dos dados coletados, sensores operando no modo polling devem ser requisitados pelo NCAP previamente. Para isto, o usuário final deverá estabelecer uma sessão SIP com o NCAP, endereçando o TransducerChannel desejado. Caso o sensor requisitado esteja conectado ao NCAP, este deverá enviar uma mensagem ao *mote*, *Start Read TransducerChannel data-set segment*, visando iniciar o envio de dados pelo sensor. Quando o usuário SIP não desejar mais receber dados sensoriados, o mesmo deverá finalizar a sessão aberta, através da requisição SIP *BYE*. Ao receber este método, o NCAP encarrega-se de enviar uma mensagem ao TransducerChannel, *Stop Read TransducerChannel data-set segment*, a fim de finalizar o fluxo vindo do sensor. Todas as mensagens contendo as informações sensoriadas encaminhadas pelo *mote* ao NCAP estarão no formato da norma IEEE 1451, correspondendo ao comando *Read TransducerChannel data-set segment*, sendo o NCAP responsável por adaptá-las às características de tempo real presentes no protocolo RTP, tais como a inclusão do número de sequência (campo *sequence number*) e da informação de tempo de entrega de cada fragmento (campo *timestamp*), e encaminhar ao SIP phone correspondente. A figura 38 apresenta as mensagens trocadas entre TIM, NCAP e SIP phone a fim de reportar dados coletados por sensores multimídia ao usuário SIP, utilizando o modo de operação polling.

Ao executar no modo TIM-IM, o TransducerChannel que desejar reportar os dados coletados deverá, primeiramente, enviar a mensagem IEEE 1451 *Start Read TransducerChannel data-set segment* ao NCAP. Ao receber esta mensagem, o NCAP tentará estabelecer uma sessão SIP com os usuários previamente registrados àquele sensor por meio do esquema de notificação do estado de conectividade, através da requisição *SUBSCRIBE*. Neste momento, o *mote* deverá transmitir os dados coletados utilizando o comando *Read TransducerChannel data-set segment* ao NCAP, o qual deverá, caso a sessão SIP tenha sido estabelecida com sucesso, adaptar as mensagens ao formato RTP e enviá-las ao SIP phone. Quando o TransducerChannel não desejar mais reportar informações, o mesmo enviará a requisição *Stop Read TransducerChannel data-set segment*, a qual corresponderá a uma mensagem *BYE* ao usuário final, finalizando a sessão SIP. Esta troca de mensagens é apresentada pelo diagrama de sequência da figura 39.

Independentemente de a sessão SIP ter sido estabelecida ou não entre o NCAP e o usuário SIP, o sensor, reportando dados através do modo TIM-IM, enviará as informações sensoriadas após a transmissão da mensagem *Start*

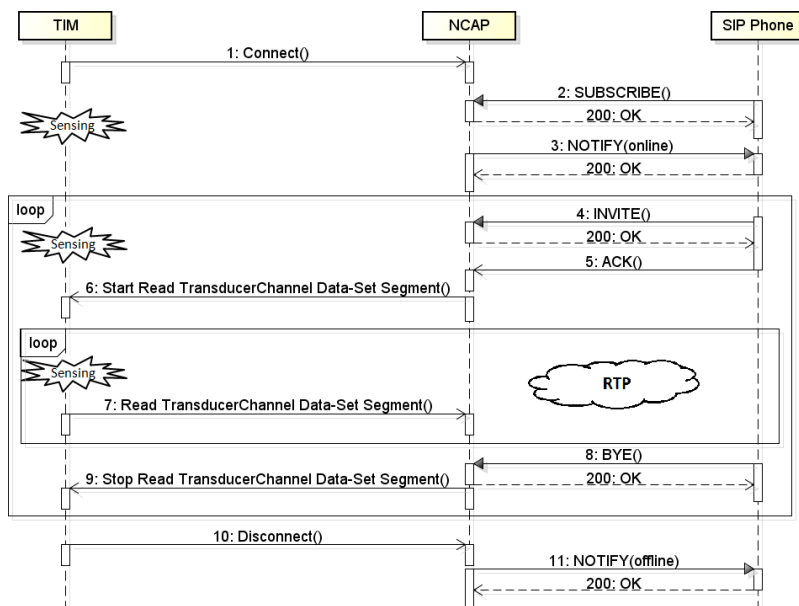


Figura 38: Diagrama de sequência para sensores multimídia utilizando o modo polling.

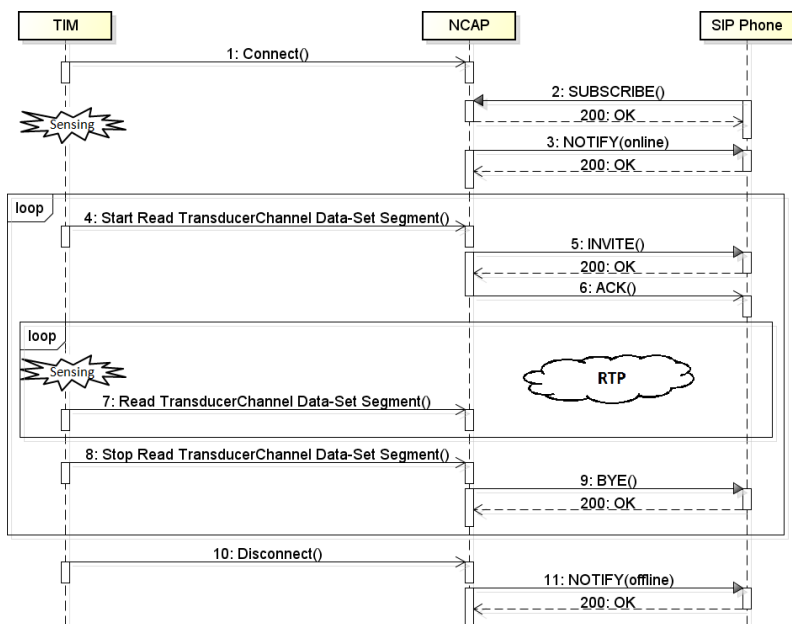


Figura 39: Diagrama de sequência para sensores multimídia utilizando o modo TIM-IM.

Read TransducerChannel data-set segment. Isto é desejável, pois, apesar de o diálogo SIP, possivelmente, não estar ativo, o NCAP poderá ser programado para tentar estabelecer uma sessão posteriormente, enviando os dados coletados, os quais foram armazenados nele. Esta situação é interessante, por exemplo, quando o sensor trata-se de uma câmera de segurança que, ao detectar movimento, tentará reportar as imagens ao dono da residência através do NCAP. Porém, caso o morador não possa atender naquele momento, o NCAP poderá enviar as informações algum tempo depois.

Uma tentativa de inicialização de sessão SIP pode ser rejeitada por basicamente dois motivos. O primeiro, por rejeição explícita do usuário sendo chamado, neste caso, enviando uma resposta SIP no formato 3xx, 4xx, 5xx ou 6xx, conforme apresentado na tabela 4 da seção 4.4. O segundo motivo representa o não envio de uma resposta final à parte chamadora, assim, causando o envio de uma requisição *CANCEL* a fim de cancelar a tentativa de estabelecimento da chamada. Esta última situação é apresentada na figura 40, onde o usuário SIP somente envia uma resposta provisória, informando que seu SIP phone está tocando, porém, a chamada não é atendida nem rejeitada a tempo.

6.4.4 Operações sobre TEDS

A norma IEEE 1451.0 (IEEE 1451.0, 2007) especifica quatro possíveis operações realizadas sobre os TEDS, as quais se enquadram na classe de comando chamada “comandos comuns ao TIM e TransducerChannel”. Estes comandos são responsáveis por: questionar (*Query TEDS*), ler (*Read TEDS segment*), escrever (*Write TEDS segment*) e atualizar (*Update TEDS*) os TEDS.

O presente trabalho visa, também, tratar estes quatro tipos de manipulações sobre os TEDS através do usuário SIP. Para isto, foi utilizado a requisição *MESSAGE*, a mesma empregada na leitura dos dados não-multi-mídia sensorizados, conforme apresentado na seção 6.4.2. Assim, estas novas mensagens possuirão um diagrama de sequência muito semelhante ao apresentado pela figura 36, porém, tanto as requisições *MESSAGE* quanto o comando de leitura enviado pelo NCAP ao TIM possuirão o conteúdo diferente, relatando a operação desejada sobre o TEDS.

O primeiro comando, chamado *Query TEDS*, é utilizado pelo NCAP para solicitar informações sobre um determinado TEDS, a fim de, posteriormente, executar um comando de leitura ou escrita sobre o mesmo. Este comando possui um único argumento numérico, o qual identifica o código do TEDS que deverá ser acessado. Assim, este trabalho utiliza o envio do

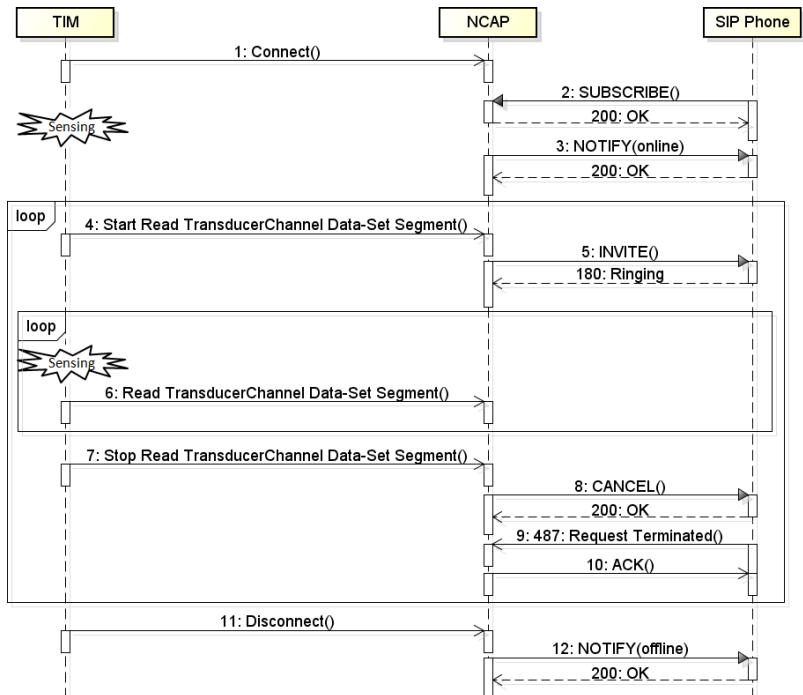


Figura 40: Diagrama de sequência para cancelamento da chamada em sensores multimídia utilizando o modo TIM-IM.

método SIP *MESSAGE* contendo em seu corpo a seguinte informação: *Query TEDS <TIM_or_TransducerChannel> <TEDSAccessCode>*. O argumento *<TIM_or_TransducerChannel>* deve ser substituído pelo valor 0 ou 1, onde 0 significa que o TEDS requisitado pertence ao TIM e 1 indica que o mesmo pertence ao TransducerChannel endereçado pela mensagem SIP. Esta distinção é importante, pois, em alguns casos, um mesmo tipo de TEDS pode ser implementado tanto pelo TIM quanto pelo TransducerChannel. O segundo parâmetro, *<TEDSAccessCode>*, deve ser preenchido com a representação numérica do TEDS desejado, conforme a tabela 17 descrita pelo documento IEEE 1451.0 (IEEE 1451.0, 2007). Ao receber uma mensagem SIP contendo este formato, o NCAP possuirá todas as informações necessárias para encaminhar o comando de questionamento ao TIM ou TransducerChannel destino. Este comando deve gerar uma resposta por parte do sensor, a qual conterá diversas informações, tais como a indicação de suporte ou não ao TEDS referenciado, se o mesmo é modificável, se o conteúdo é válido, seu tamanho, entre outras. Desta forma, ao receber a resposta, o NCAP deverá encaminhar a mesma ao usuário SIP ao qual requisitou previamente. Isto será realizado através de uma nova requisição *MESSAGE* do protocolo SIP, contendo a string *Query TEDS Reply TEDSAttrib=<p1> TEDSStatus=<p2> TEDSSize=<p3> TEDSCkSum=<p4> MaxTEDSSize=<p5>*, sendo que p1 até p5 representam campos de parâmetros e devem ser preenchidos com os mesmos valores presentes na resposta enviada pelo sensor ao NCAP, em formato texto.

O segundo comando tratado é conhecido pela norma como *Read TEDS segment*, o qual possui dois parâmetros. Como o nome sugere, o mesmo executa a leitura de um determinado segmento de um TEDS, cujo código de acesso é indicado em seu primeiro argumento. O segundo representa um campo de deslocamento, indicando o endereço relativo ao início do TEDS no qual o bloco de dados deve ser lido. Desta forma, deve-se enviar o método SIP *MESSAGE*, utilizando o seguinte formato em seu corpo: *Read TEDS segment <TIM_or_TransducerChannel> <TEDSAccessCode> <TEDSOffset>*, onde os dois primeiros parâmetros devem ser preenchidos conforme indicado pelo comando anterior, *Query TEDS*, enquanto que o campo *<TEDSOffset>* indicará o deslocamento da leitura. A resposta enviada pelo NCAP ao SIP phone deverá conter a informação *Read TEDS segment Reply <TEDSOffset> <RawTEDSBlock>*, com o campo de deslocamento e os blocos do TEDS requisitado preenchidos conforme recebido na resposta originada pelo TIM ou TransducerChannel. Como explicado na seção 6.4.2, a mensagem SIP utilizada envia as informações em formato texto, enquanto que o padrão IEEE 1451 utiliza os octetos em formato numérico. Assim, o NCAP é responsável por realizar as transformações necessárias antes de repassar as mensagens

tanto para o TIM quanto para o SIP phone, evitando possíveis problemas na apresentação dos valores.

O próximo comando, *Write TEDS segment*, é utilizado para escrever uma parte de um determinado TEDS. Seus argumentos, definidos pela norma, são: o código de acesso do TEDS desejado, o deslocamento utilizado para identificar em que parte do TEDS os dados serão escritos e, por último, o conteúdo que será escrito. Assim, seguindo o mesmo formato apresentado pelos outros comandos, é utilizado o método *MESSAGE* contendo o seguinte formato: *Write TEDS segment <TIM_or_TransducerChannel> <TEDSAccessCode> <TEDSOffset> <RawTEDSBlock>*. Este comando não gera nenhuma resposta por parte do sensor, consequentemente o usuário SIP também não receberá retorno.

Quando o TIM começa a sobrescrever um TEDS existente através do comando *Write TEDS segment*, o TEDS sendo escrito deve ser marcado como inválido. O mesmo deve permanecer neste estado até que o comando *Update TEDS* seja recebido. Ao recebê-lo, o TIM deverá copiar o TEDS para a memória não-volátil, se possível, marcando o mesmo como válido novamente. Segundo a norma IEEE 1451.0, este comando possui somente um argumento, representando o código de acesso do TEDS. Desta forma, o método *MESSAGE* será composto pelo seguinte formato: *Update TEDS <TIM_or_TransducerChannel> <TEDSAccessCode>*. A norma especifica que a resposta a este comando, enviada pelo sensor, deverá conter a mesma informação apresentada na resposta ao comando *Query TEDS*. Logo, o usuário final, através de seu SIP phone, receberá uma mensagem SIP similar a apresentada previamente pela resposta ao comando *Query TEDS*, ou seja: *Update TEDS Reply TEDSAttrib=<p1> TEDSStatus=<p2> TEDSSize=<p3> TEDSCkSum=<p4> MaxTEDSSize=<p5>*.

6.4.5 Demonstrações e Resultados

A seguir serão apresentadas algumas imagens que demonstram as operações de leitura dos dados sensorizados pelos *motes*. O cenário consiste de dois dispositivos sensores EPOSMoteII, um coletando a temperatura ambiente e o outro, um sensor multimídia, capturando o áudio circundante através de seu microfone, sendo os mesmos utilizados para os testes das seções 6.2 e 6.3. O NCAP é implementado na linguagem de programação C++ e roda em um computador pessoal, utilizando o sistema operacional Linux. Por fim, o usuário final utiliza outro computador, também com o ambiente Linux, conectado à mesma rede interna do NCAP e executando o SIP phone Jitsi.

A figura 41 apresenta a leitura das amostras de temperatura obtidas

pelo sensor escalar, o qual opera no modo polling. Assim, a primeira mensagem da caixa de diálogo da imagem mostra o comando enviado pelo usuário ao NCAP, o qual está no formato *Read TransducerChannel data-set segment <DataSetOffset>*. Esta mensagem será adicionada no corpo de uma requisição SIP *MESSAGE*, conforme apresentado abaixo:

```
MESSAGE sip:temperature_sensor_1@10.0.0.102 SIP/2.0
Call-ID: 4ef296963edd51a97446d63a67414d0f@0:0:0:0:0:0:0:0
CSeq: 26057406 MESSAGE
From: 'Teste' <sip:Teste@10.0.0.106:5060;transport=udp>;tag=48cf
d0e7
To: <sip:temperature_sensor_1@10.0.0.102>
Via: SIP/2.0/UDP 10.0.0.106:5060;branch=z9hG4bK-3336-7f53ed12e8af7
55de88e1f9281618a80
Max-Forwards: 70
Content-Type: text/plain
Contact: 'Teste' <sip:Teste@10.0.0.106:5060;transport=udp>
User-Agent: Jitsi1.0-beta1-nightly.build.3820Linux
Content-Length: 41
```

```
Read TransducerChannel data-set segment 0
```

O método SIP será transformado em um comando no formato IEEE 1451 e enviado ao sensor, o qual retornará os dados coletados no mesmo formato. Ao receber a resposta, o NCAP criará uma nova requisição SIP *MESSAGE*, e enviará ao SIP phone que havia solicitado a informação. A segunda mensagem da figura 41 apresenta a resposta recebida pelo usuário final, a qual está no formato descrito pelo presente trabalho: *Read TransducerChannel data-set segment Reply <ReadSensorOffset> <ReadSensorData>*. A seguir, é apresentado o método *MESSAGE* enviado pelo NCAP ao SIP phone, contendo 10 amostras de temperatura indicando 23 e 22°C.

```
MESSAGE sip:Teste@10.0.0.106:5060 SIP/2.0
Via: SIP/2.0/UDP 10.0.0.102:5060;branch=z9hG4bK1418418679
To: <sip:Teste@10.0.0.106:5060>
From: <sip:temperature_sensor_1@10.0.0.102>;tag=824479389
Call-ID: 218731730@10.0.0.102
CSeq: 1 MESSAGE
Max-Forwards: 70
Contact: <sip:temperature_sensor_1@10.0.0.102>
Content-Type: text/plain
Content-Length: 77
```

```
Read TransducerChannel data-set segment Reply 0 23,23,22,22,22,22,
22,22,22,22
```

A figura 42 apresenta uma sessão estabelecida entre o usuário SIP e o NCAP, a fim de receber as amostras de áudio obtidas pelo sensor multimídia.

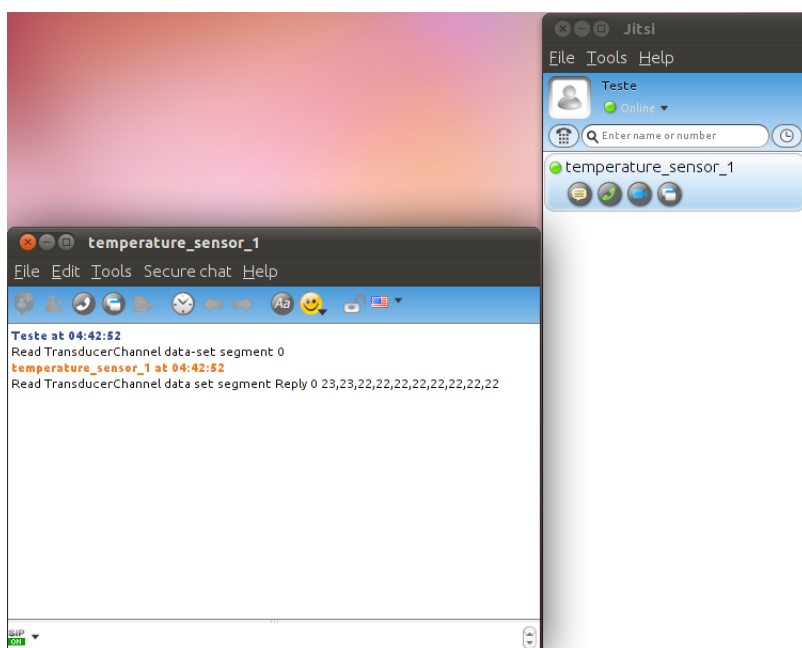


Figura 41: SIP phone Jitsi conectado a um sensor de temperatura.

A requisição *INVITE*, a qual inicializa o diálogo, é apresentada abaixo. O protocolo SDP, presente no corpo da mensagem SIP, descreve que a sessão será estabelecida utilizando o codec G.711 A-law, preferivelmente, o qual é bastante utilizado junto ao VoIP. Assim, como o *mote* envia os dados coletados utilizando o codec G.726, como comentado na seção 6.3.2, o NCAP deve executar as conversões necessárias.

```
INVITE sip:audio_sensor_1@10.0.0.102 SIP/2.0
Call-ID: a00d4ffb9c251dd8866fd49281599a73@0:0:0:0:0:0
CSeq: 1 INVITE
From: 'Teste' <sip:Teste@10.0.0.106:5060;transport=udp>;tag=3bfd7578
To: <sip:audio_sensor_1@10.0.0.102>
Via: SIP/2.0/UDP 10.0.0.106:5060;branch=z9hG4bK-3336-0760de683ff0e5065c84926853c44100
Max-Forwards: 70
Contact: 'Teste' <sip:Teste@10.0.0.106:5060;transport=udp>
User-Agent: Jitsi1.0-beta1-nightly.build.3820Linux
Content-Type: application/sdp
Content-Length: 681
```

```
v=0
o=Teste 0 0 IN IP4 10.0.0.106
s=-
c=IN IP4 10.0.0.106
t=0 0
m=audio 5012 RTP/AVP 8 0 101
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:101 telephone-event/8000
```

É interessante que sensores multimídia autônomos operando através do modo TIM-IM, tal como o microfone aqui utilizado, possua um detector de atividade de voz a fim de desconectar-se frente ao NCAP quando for detectado silêncio, ou seja, quando não houver informação útil a ser transferida. Assim, foi analisado o tempo de estabelecimento e término da conexão do *mote*, levando em consideração as três partes do sistema (TIM, NCAP e SIP phone), a fim de verificar se é mais vantajoso que o sensor permaneça enviando silêncio ou desconecte-se do NCAP, permitindo desligar o rádio e economizar a bateria e a banda de rede, e reconecte-se novamente posteriormente. Através dos diagramas de sequência das figuras 31 e 39, percebe-se que antes de iniciar o fluxo de dados no modo TIM-IM, há a conexão do TIM frente ao NCAP (feito através de uma conexão TCP tradicional), seguido pelo comando *TransducerChannel Operate* (mensagem TCP). O NCAP é responsável por enviar uma requisição *NOTIFY* ao usuário SIP, informando que o TransducerChannel está *on-line*, recebendo uma resposta 200 OK. Neste mo-

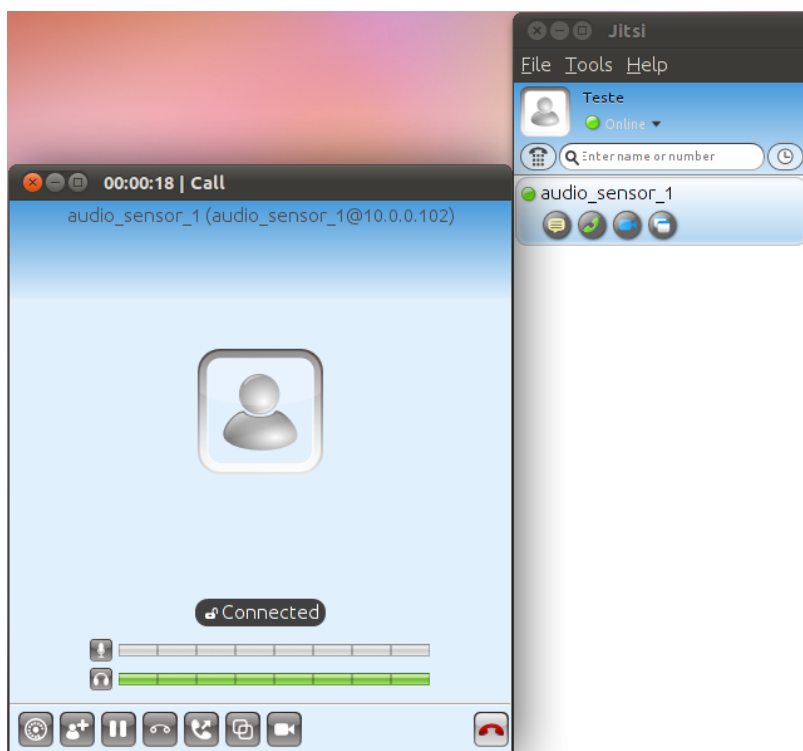


Figura 42: SIP phone Jitsi conectado a um sensor de áudio.

mento, o TIM envia um comando *Start Read TransducerChannel data-set segment* no formato IEEE 1451, fazendo com que o NCAP estabeleça uma sessão SIP utilizando os métodos *INVITE*, *ACK* e a resposta 200 OK. Este processo, considerando que a resposta do usuário SIP seja imediata, leva 159 milissegundos. Já a desconexão é realizada pela mensagem IEEE 1451 *Stop Read TransducerChannel data-set segment*, seguida pelo método SIP *BYE* e a resposta 200 OK. Após isto, há a desconexão TCP do sensor, causando o envio de uma mensagem *NOTIFY* ao usuário SIP, contendo em seu conteúdo a informação de que o sensor está *off-line*, retornando outro 200 OK. Este processo de desconexão leva 412 milissegundos para ser executado. Logo, são gastos 0,571 segundos para iniciar e finalizar a conexão de um *mote*, o qual executa no modo TIM-IM. Assim, nos casos onde o período de silêncio do sensor é maior que este valor, é mais lucrativo executar o processo de conexão e desconexão quando não há informações significativas para enviar, ao invés de manter a conexão e enviar silêncio, permitindo que o sensor economize sua energia e a banda de rede.

Outro fator importante além do tempo de conexão e desconexão, é o tamanho dos pacotes enviados durante estes processos. Então, foi calculado a quantidade de bytes enviados entre TIM, NCAP e SIP phone, considerando somente as informações referentes à norma IEEE 1451 e aos protocolos SIP e RTP, desconsiderando os dados das camadas inferiores (TCP/IP e UDP/IP). Assim, para a conexão do *mote*, há o envio de 2.487 bytes¹ (referente às mensagens IEEE 1451 *TransducerChannel Operate*, SIP *NOTIFY* e 200 OK, IEEE 1451 *Start Read TransducerChannel data-set segment*, SIP *INVITE*, *ACK* e 200 OK). Já para a desconexão, são enviados 1.708 bytes¹ (através das mensagens IEEE 1451 *Stop Read TransducerChannel data-set segment*, SIP *BYE* e 200 OK, SIP *NOTIFY* e 200 OK). Ou seja, totalizando 4.195 bytes para a conexão e desconexão do sensor. Como discutido na seção 6.3.2, cada pacote de áudio enviado pelo *mote*, utilizando o codec G.726, contém 10 bytes referentes à norma IEEE 1451 e 73 bytes de áudio, sendo enviados 27 pacotes por segundo, ou seja, 2.241 bytes por segundo. O áudio que chega ao NCAP, partindo do TIM, é enviado para o usuário SIP utilizando o codec G.711 A-law. Assim, são enviados 50 pacotes por segundo, cada um contendo 172 bytes (160 referentes ao áudio e 12 ao protocolo RTP), totalizando 8.600 bytes por segundo (10.841 bytes por segundo levando em consideração ambos os lados do NCAP). Logo, são utilizados 4.195 bytes para que o *TransducerChannel* conecte-se e desconecte-se ao NCAP e 10.841 bytes para o envio de cada segundo de áudio. Assim, prova-se ser mais vantajoso finalizar a conexão quando utilizado TIM-IM, ao invés de enviar silêncio. Os valores

¹ Este valor pode variar, pois o protocolo SIP é descrito em formato texto, logo, as mensagens podem conter diferentes tamanhos em SIP phones e situações distintas.

aqui apresentados são resumidos na tabela 12.

Tabela 12: Tempo e número de bytes trafegados para a conexão e desconexão de um sensor TIM-IM.

	Conexão	Desconexão	Total
Tempo (ms)	159	412	571
Tamanho (bytes)	2.487	1.708	4.195

Este capítulo deixou claro que apesar de o padrão IEEE 1451 introduzir diversos benefícios, o mesmo deixa de lado alguns fatores importantes relacionados a redes de sensores sem fio. Assim, primeiramente mostrou-se ser possível um sensor operar de forma autônoma de acordo com a norma, através do modo de execução TIM-IM, poupando uma quantidade significativa de sua fonte de energia comparado ao modo polling. Também, foi necessário adaptar a norma visando executar sensores multimídia, os quais são ignorados pela mesma. Com a introdução desta nova classe de sensores, os métodos de acesso a transdutores inteligentes descritos pela norma, IEEE 1451.1 e IEEE 1451.0 HTTP, tornam-se inutilizáveis. Logo, é apresentada a integração do IEEE 1451 ao protocolo SIP, permitindo que sensores escalares sejam acessados utilizando mensagens instantâneas, enquanto que sensores multimídia utilizam o estabelecimento de sessões SIP. Desta forma, usuários utilizando SIP phones podem acessar uma infinidade de sensores remotos através da Internet, independentemente de sua localização física.

7 CONCLUSÃO

Redes de sensores sem fio representam uma tecnologia bastante popular atualmente. Estas redes são compostas por dispositivos chamados nós sensores, os quais são capazes de monitorar algum fenômeno físico, biológico ou químico do meio ambiente que os rodeia, transformando-os em sinais digitais e comunicando-se com outros nós da rede, muitas vezes através de um rádio de baixa potência. RSSF são utilizadas em inúmeros cenários, tais como na área militar e industrial, para monitoramento do meio ambiente e tráfego, em casas inteligentes, entre outros. Os dispositivos sensores podem coletar informações escalares, tais como temperatura, aceleração, luz e etc., ou podem capturar fluxos de dados, como imagens, vídeos e áudios, sendo assim, chamados de sensores multimídia.

Com o crescente uso das RSSF surgiram diversas plataformas, tanto de hardware como de software, visando a utilização em nós sensores. Desta forma, a fim de padronizar o acesso e o comportamento dos *nodes*, a família de padrões IEEE 1451 foi desenvolvida. Esta padronização introduz conceitos interessantes ao mundo dos transdutores, tais como a divisão do sistema em duas partes principais, NCAP e TIM, e a definição dos TEDS, os quais representam dispositivos de memória anexados ao próprio transdutor, armazenando informações sobre o mesmo e possibilitando sua auto-identificação em frente ao sistema.

Por focar principalmente em interoperabilidade, o padrão IEEE 1451 não trata eficientemente os requisitos das redes de sensores sem fio atuais, tal como a necessidade dos sensores executarem autonomamente, permitindo a hibernação dos mesmos a fim de economizar sua fonte de energia. Assim, este trabalho deu continuidade à proposta de Torri (TORRI, 2008), onde através do modo de execução TIM-IM, WTIMs podem reportar seus dados sempre que houver novas leituras sensorizadas, sem a necessidade de polling por parte do NCAP. Resultados mostraram que o método TIM-IM é capaz de reduzir consideravelmente o consumo de energia comparado ao método polling. Após 1 hora de testes, o sensor operando no modo TIM-IM consumiu menos da metade da quantidade de energia comparado ao *node* rodando no modo polling (129,83 J para o TIM-IM otimizado, 156,13 J para o TIM-IM e 316,96 J para o polling), podendo obter resultados ainda melhores em outros cenários de teste.

Adicionalmente, o padrão IEEE 1451 limita-se às redes de sensores que captam informações escalares, não tratando sensores multimídia. Assim, a presente dissertação visa, também, a integração deste tipo de dispositivo à norma. Logo, são apresentadas algumas modificações tanto nos TEDS quanto

nas mensagens trafegadas entre NCAP e TIM, não afetando o comportamento atual da norma, somente representando uma extensão à mesma. A fim de permitir o acesso a sensores multimídia, além dos escalares, através da rede do usuário, nada mais justo do que a utilização de um protocolo próprio para o envio de fluxos de dados. Assim, foi utilizado o protocolo SIP.

SIP é um protocolo o qual vem sendo muito utilizado atualmente junto à tecnologia VoIP, sendo responsável por estabelecer, modificar e finalizar uma sessão, tal como chamadas de voz e/ou vídeo. Visando obter comunicação com entidades SIP remotas, o mesmo utiliza o protocolo IP para a camada de rede e, normalmente, os protocolos TCP ou UDP na camada de transporte, estando localizado na camada de sessão. O protocolo SIP representa, somente, o meio de manipulação de sessões, trabalhando em conjunto com outros protocolos a fim de obter o completo funcionamento do sistema VoIP. Por exemplo, o protocolo SDP é utilizado no corpo das mensagens SIP a fim de descrever os parâmetros da sessão que visa ser estabelecida e RTP é responsável pelo envio e recebimento dos dados que representam os fluxos de informações propriamente dito.

Devido ao fato de o protocolo SIP possuir um tamanho relativamente grande e, sendo assim, não ser viável em muitos sistemas embarcados com restrição de recursos, este trabalho apresentou uma miniaturização do mesmo. Esta adaptação foi alcançada através da eliminação de algumas requisições e campos de cabeçalho do protocolo SIP, entretanto, não comprometendo a funcionalidade do mesmo, sendo compatível com a norma original. Resultados mostraram que, com todas as reduções feitas, foi obtida uma biblioteca final com 87.450 bytes, quase três vezes menor do que o SIP library (237.306 bytes) e oito vezes menor do que a parte referente ao SIP do Asterisk (685.562 bytes).

Por fim, o presente trabalho apresentou a integração do protocolo SIP ao padrão IEEE 1451 para redes de sensores sem fio. Para isto, foi utilizado o esquema de notificação de presença presente no SIP, permitindo reconhecer o estado de conectividade de um sensor, sabendo se o mesmo está conectado ou não ao NCAP. Também, utilizou-se a extensão relativa à transferência de mensagens instantâneas, permitindo que sensores não-multimídia reportem seus dados sensorizados, sem a necessidade de estabelecer uma sessão SIP, como ocorre para os sensores os quais enviam fluxos de informações. O envio de mensagens instantâneas permitiu, também, a manipulação dos TEDS pelo usuário remoto. Para finalizar, foi verificado o tempo e a quantidade de bytes trafegados para o estabelecimento e término da conexão do sensor, o qual opera no modo TIM-IM. Assim, a conexão, envolvendo as três partes do sistema (TIM, NCAP e SIP phone), leva 159 milissegundos para ser executada e são enviados 2.487 bytes, enquanto que a desconexão dura 412

milissegundos e trafega 1.708 bytes, totalizando 0,571 segundos e 4.195 bytes. Como o envio de cada segundo de áudio necessita de 10.841 bytes e o tempo de conexão e desconexão é pequeno, mostra-se ser mais lucrativo finalizar a conexão quando utilizado TIM-IM ao invés de enviar silêncio, permitindo desligar o rádio e economizar energia e a banda de rede.

Assim, com a união do protocolo SIP ao IEEE 1451, permite-se que sensores sejam acessados por usuários remotos utilizando SIP phones através da Internet, independentemente de sua localização física. Esta abordagem pode ser utilizada em uma diversidade de cenários e parece ser bastante promissora quando integrado ao Internet das Coisas, onde pessoas poderão interagir através de seu ambiente virtual, utilizando o protocolo SIP, a uma diversidade de objetos do mundo real, independentemente de seu tamanho físico.

7.1 TRABALHOS FUTUROS

A seção 6.1 apresentou uma versão miniatura do protocolo SIP a fim de executá-la em sistemas embarcados com restrição de memória. Porém, outros fatores podem ser considerados, tal como a utilização da banda de rede. Assim, novas análises podem ser feitas a fim de minimizar o consumo da rede, por exemplo, o uso da forma compacta dos cabeçalhos SIP, conforme descrito pela norma. Também, pesquisas relativas ao protocolo RTP seriam interessantes, pois a quantidade de mensagens SIP enviadas representa um número bastante inferior comparado aos pacotes RTP, os quais trafegam os fluxos de informações.

Quanto às manipulações sobre os TEDS através do usuário SIP, apresentadas na seção 6.4.4, pode-se pensar em outras abordagens. O padrão IEEE 1451 descreve a utilização de esquemas XML para a descrição de alguns TEDS. Desta forma, é possível fazer as requisições SIP *MESSAGE* enviar os TEDS no formato XML, ao invés de uma sequência de valores numéricos como apresentado nesta dissertação. Porém, ambas as abordagens possuem suas vantagens e desvantagens, por exemplo, o formato numérico representa uma forma compacta de enviar os TEDS, enquanto que esquemas XML ocupariam um espaço muito maior. Por outro lado, a utilização do formato XML simplificaria a leitura dos TEDS para o usuário final comparado à sequência de valores numéricos, pois, para este último, deve ser feita uma análise detalhada de cada valor, utilizando as tabelas que descrevem as estruturas dos TEDS na norma.

REFERÊNCIAS BIBLIOGRÁFICAS

AKYILDIZ, I.; MELODIA, T.; CHOWDURY, K. Wireless multimedia sensor networks: A survey. *Wireless Communications, IEEE*, v. 14, n. 6, p. 32–39, December 2007. ISSN 1536-1284.

AKYILDIZ, I. F. et al. A Survey on Sensor Networks. In: *IEEE Communications Magazine*. [S.l.: s.n.], 2002. v. 40, n. 8, p. 102–114.

ALNUAIMI, M.; SALLABI, F.; SHUAIB, K. A survey of wireless multimedia sensor networks challenges and solutions. In: *Innovations in Information Technology (IIT), 2011 International Conference on*. [S.l.: s.n.], 2011. p. 191–196.

BERISHA, V.; KWON, H.; SPANIAS, A. Real-time acoustic monitoring using wireless sensor motes. In: *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*. [S.l.: s.n.], 2006. p. 847–850.

CAMARILLO, G. *SIP Demystified*. First. [S.l.]: McGraw-Hill Professional, 2001. 320 p. ISBN 0071373403.

CAMPBELL, B. et al. *Session Initiation Protocol (SIP) Extension for Instant Messaging. RFC 3428*. [S.l.], December 2002. 19 p.

CHENG, B.-C.; CHEN, H.; TSENG, R.-Y. Context-Aware Gateway for Ubiquitous SIP-Based Services in Smart Homes. In: *International Conference on Hybrid Information Technology*. Cheju Island: [s.n.], 2006. v. 2, p. 374–381.

CHUNSHAN, S. et al. Application of session initiation protocol to networked sensor interfaces. *Computer Standards & Interfaces*, Amsterdam, The Netherlands, The Netherlands, v. 31, p. 454–457, February 2009.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. *Sistemas Distribuídos - Conceitos e Projeto*. BOOKMAN COMPANHIA ED, 2007. ISBN 9788560031498. <<http://books.google.com/books?id=KSZ1rRWmUoC>>.

DIGIUM. *Asterisk – The Open Source Telephony Projects*. 1999. Website. Acessado em 15 de julho de 2010. <<http://www.asterisk.org/>>.

DONOVAN, S. *The SIP INFO Method. RFC 2976*. [S.l.], October 2000. 10 p.

ELSON, J.; ESTRIN, D. *An Address-Free Architecture for Dynamic Sensor Networks*. University of Southern California, Computer Science Department, 2000. Technical Report 00-724.

ENGBLOM, J. *Worst-Case Execution Time Analysis*. IAR Systems/Uppsala University: [s.n.], 2002.

ESWARAN, A.; ROWE, A.; RAJKUMAR, R. Nano-rk: an energy-aware resource-centric rtos for sensor networks. In: *Real-Time Systems Symposium, 2005. RTSS 2005. 26th IEEE International*. [S.l.: s.n.], 2005. p. 10 pp. –265. ISSN 1052-8725.

FRÖHLICH, A. A. *Application-Oriented Operating Systems*. Sankt Augustin: GMD - Forschungszentrum Informationstechnik, 2001. 200 p. ISBN 3-88457-400-0. <<http://www.lisha.ufsc.br/guto/publications/aos.pdf>>.

FRÖHLICH, A. A.; WANNER, L. F. Operating System Support for Wireless Sensor Networks. *Journal of Computer Science*, v. 4, n. 4, p. 272–281, 2008.

GILSINN, J. D.; LEE, K. Wireless interfaces for ieee 1451 sensor networks. In: *Proceedings of the First ISA/IEEE Conference Sensor for Industry*. Rosemont, IL , USA: [s.n.], 2001. p. 45–50. ISBN 0-7803-6659-X.

GOSZTOLYA, G.; PACZOLAY, D.; ANDTH, L. T. Low-complexity audio compression methods for wireless sensors. In: *Intelligent Systems and Informatics (SISY), 2010 8th International Symposium on*. [S.l.: s.n.], 2010. p. 77–81.

HANDLEY, M.; JACOBSON, V.; PERKINS, C. *SDP: Session Description Protocol. RFC 4566*. [S.l.], July 2006. 50 p.

HEINZELMAN, W. B. *Application-Specific Protocol Architectures for Wireless Networks*. Tese (Doutorado) — Massachusetts Institute of Technology, June 2000.

HILL, J. et al. System architecture directions for networked sensors. *SIGPLAN Not.*, ACM, New York, NY, USA, v. 35, p. 93–104, November 2000. ISSN 0362-1340.

IEEE 1451.0. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats*. [S.l.], September 2007. 335 p.

IEEE 1451.1. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor (NCAP) Information Model*. [S.l.], 2000. 349 p.

IEEE 1451.2. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*. [S.l.], 1998. 120 p.

IEEE 1451.3. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems*. [S.l.], 2004. 175 p.

IEEE 1451.4. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*. [S.l.], 2004. 430 p.

IEEE 1451.5. *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*. [S.l.], October 2007. 236 p.

IEEE 1451.6. *Proposed Standard for a High-Speed CANopen-Based Transducer Network Interface for Intrinsically Safe and Non-Intrinsically Safe Applications*. [S.l.]. <<http://grouper.ieee.org/groups/1451/6/>>.

IEEE 1451.7. *IEEE Standard for Smart Transducer Interface for Sensors and Actuators - Transducers to Radio Frequency Identification (RFID) Systems Communication Protocols and Transducer Electronic Data Sheet Formats*. [S.l.], June 2010. 99 p.

International Telegraph And Telephone Consultative Committee. 40, 32, 24, 16 kbit/s *Adaptive Differential Pulse Code Modulation (ADPCM)*. Recommendation G.726. [S.l.], 1990. 59 p.

JOHNSTON, A. B. *SIP: Understanding the Session Initiation Protocol*. Second. [S.l.]: Artech House Publishers, 2003. 310 p. ISBN 1580536557.

KWAK, J.-Y. Ubiquitous Services System Based on SIP. In: *IEEE Transactions on Consumer Electronics*. Rosemont, IL, USA: [s.n.], 2007. v. 53, n. 3, p. 938–944.

LAKAY, E. T.; AGBINYA, J. I. SIP-Based Content Development for Wireless Mobile Devices. In: *1st International Conference on Computers, Communications, e Signal Processing with Special Track on Biomedical Engineering*. Kuala Lumpur, Malaysia: [s.n.], 2005. p. 130–134. ISBN 978-1-4244-0011-9.

LANGENDOEN, K.; HALKES, G. *Embedded Systems Handbook*. [S.l.]: CRC Press, 2005. 1160 p.

LEE, K. A Synopsis of the IEEE P1451- Standards for Smart Transducer Communication. In: . National Institute of Standards and Technology: [s.n.]. <ieee1451.nist.gov/1451synopsis-599F.pdf>.

LEE, K. IEEE 1451: A Standard in Support of Smart Transducer Networking. In: *17th IEEE Instrumentation and Measurement Technology Conference*. Baltimore, MD, USA: [s.n.], 2000. v. 2, p. 525–528. ISBN 0-7803-5890-2.

LEE, K.; SONG, E. Object-oriented application framework for ieee 1451.1 standard. *IEEE Transactions on Instrumentation and Measurement*, v. 54, n. 4, p. 1527–1533, August 2005. ISSN 0018-9456.

LEE, K.; SONG, E. Wireless Sensor Network Based on IEEE 1451.0 and IEEE 1451.5-802.11. In: *8th International Conference on Electronic Measurement and Instruments*. [S.l.: s.n.], 2007. p. 4–7.

LEE, K. B.; SONG, E. Y. Sensor Alert Web Service for IEEE 1451-Based Sensor Networks. In: *Instrumentation and Measurement Technology Conference*. Singapore: [s.n.], 2009. p. 1049–1054. ISBN 978-1-4244-3352-0.

LIAO, J. et al. A Demand-Driven Parsing Method for SIP Offload in Home Network. In: *IEEE Transactions on Consumer Electronics*. [S.l.: s.n.], 2009. v. 55, n. 3, p. 1308–1314.

LIN, A. D. et al. Wireless Integrated Network Sensors: Low Power Systems on a Chip. In: *24th European Solid-State Circuits Conference*. [S.l.: s.n.], 1998. p. 9–16.

LISHA. *EPOS Project*. 2010. Website. Acessado em 12 de novembro de 2010. <<http://epos.lisha.ufsc.br>>.

MACHADO, A. *Análise de Tempo de Execução em Alto Nível para Sistemas de Tempo Real Utilizando-se o Framework LLVM*. Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil: [s.n.], 2008.

MAINWARING, A. et al. Wireless Sensor Networks for Habitat Monitoring. In: *International Workshop on Wireless Sensor Networks and Applications*. Atlanta, Georgia, USA: [s.n.], 2002. p. 88–97. ISBN 1-58113-589-0.

MANGHARAM, R. et al. Voice over sensor networks. In: *Real-Time Systems Symposium, 2006. RTSS '06. 27th IEEE International*. [S.l.: s.n.], 2006. p. 291–302. ISSN 1052-8725.

MARCONDES, H. et al. EPOS: Um Sistema Operacional Portável para Sistemas Profundamente Embarcados. In: *III Workshop de Sistemas Operacionais*. Campo Grande, MS, Brasil: [s.n.], 2006. p. 31–45.

MOIZARD, A. *The GNU oSIP library*. 2002. Website. Acessado em 13 de julho de 2010. <<http://www.gnu.org/software/osip/>>.

OASIS Standard. *Web Services Base Notification 1.3, (WS-BaseNotification)*. [S.l.], October 2006. 68 p. <http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf>.

OASIS Standard. *Web Services Brokered Notification 1.3 (WS-BrokeredNotification)*. [S.l.], October 2006. 43 p. <http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf>.

OASIS Standard. *Web Services Topics 1.3 (WS-Topics)*. [S.l.], October 2006. 40 p. <http://docs.oasis-open.org/wsn/wsn-ws_topics-1.3-spec-os.pdf>.

OH, Y.-J.; PAIK, E.-H.; PARK, K.-R. Design of a SIP-based Real-time Visitor Communication and Door Control Architecture using a Home Gateway. In: *IEEE Transactions on Consumer Electronics*. Rosemont, IL, USA: [s.n.], 2006. v. 52, n. 4, p. 1256–1260.

OOSTDYK, R. L.; MATA, C. T.; PEROTTI, J. M. A Kennedy Space Center implementation of IEEE 1451 networked smart sensors and lessons learned. In: *IEEE Aerospace Conference*. [S.l.: s.n.], 2006. v. 2, p. 20.

ROACH, A. B. *Session Initiation Protocol (SIP)-Specific Event Notification. RFC 3265*. [S.l.], June 2002. 39 p.

RÖMER, K.; MATTERN, F. The Design Space of Wireless Sensor Networks. In: *IEEE Wireless Communications*. [S.l.: s.n.], 2004. v. 11, n. 6, p. 54–61.

ROSENBERG, J. *The Session Initiation Protocol (SIP) UPDATE Method. RFC 3311*. [S.l.], September 2002. 14 p.

ROSENBERG, J. *A Presence Event Package for the Session Initiation Protocol (SIP). RFC 3856*. [S.l.], August 2004. 28 p.

ROSENBERG, J.; SCHULZRINNE, H. *An Offer/Answer Model with the Session Description Protocol (SDP). RFC 3264*. [S.l.], June 2002. 25 p.

ROSENBERG, J.; SCHULZRINNE, H. *Reliability of Provisional Responses in the Session Initiation Protocol (SIP)*. RFC 3262. [S.l.], June 2002. 15 p.

ROSENBERG, J. et al. *SIP: Session Initiation Protocol*. RFC 3261. [S.l.], June 2002. 270 p.

ROWE, A.; MANGHARAM, R.; RAJKUMAR, R. Rt-link: A time-synchronized link protocol for energy-constrained multi-hop wireless networks. In: *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*. [S.l.: s.n.], 2006. v. 2, p. 402–411.

RUFINO, L. M. *Análise de Tempo de Execução Utilizando LLVM*. Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil: [s.n.], 2009.

SCHULZRINNE, H. et al. *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550. [S.l.], July 2003. 89 p.

SONG, E.; LEE, K. An Implementation of Smart Transducer Web Services for IEEE 1451-based Sensor Systems. In: *IEEE Sensors and Applications Symposium*. San Diego, California, USA: [s.n.], 2007. p. 1–6.

SONG, E.; LEE, K. Smart Transducer Web Services Based on the IEEE 1451.0 Standard. In: *Instrumentation and Measurement Technology Conference*. Warsaw, Poland: [s.n.], 2007. p. 1–6. ISBN 1-4244-0588-2.

SONG, E. Y.; LEE, K. An implementation of the proposed ieee 1451.0 and 1451.5 standards. In: *Proceedings of the 2006 IEEE Sensors Applications Symposium*. Rosemont, IL , USA: [s.n.], 2006. p. 72–77. ISBN 0-7803-9580-8.

SONG, E. Y.; LEE, K. Understanding IEEE 1451-Networked Smart Transducer Interface Standard - What Is a Smart Transducer? In: *IEEE Instrumentation & Measurement Magazine*. [S.l.: s.n.], 2008. v. 11, n. 2, p. 11–17.

SPARKS, R. J. *The Session Initiation Protocol (SIP) Refer Method*. RFC 3515. [S.l.], April 2003. 24 p.

STEINER, R.; MÜCK, T. R.; FRÖHLICH, A. A. A Configurable Medium Access Control Protocol for IEEE 802.15.4 Networks. In: *International Congress on Ultra Modern Telecommunications and Control Systems*. Moscow, Russia: [s.n.], 2010. p. 1–8. ISBN 978-1-4244-7284-0. <<http://www.lisha.ufsc.br/pub/icumt2010.pdf>>.

SUGANO, H. et al. *Presence Information Data Format (PIDF)*. RFC 3863. [S.l.], August 2004. 29 p.

TENNENHOUSE, D. Proactive Computing. *Communications of the ACM*, New York, NY, USA, v. 43, n. 5, p. 43–50, May 2000.

TORRI, L. B. *A Norma IEEE 1451 Aplicada a Redes Heterogêneas de Sensores Sem Fio*. Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil: [s.n.], Outubro 2008.

WANNER, L. F. *The EPOS System Supporting Wireless Sensor Networks Applications*. Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil: [s.n.], Fevereiro 2004.

WANNER, L. F. *Um Ambiente de Suporte à Execução de Aplicações em Rede de Sensores Sem Fio*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil, Dezembro 2006.