GT-TeI: Testbed para Espaços Inteligentes

RT1 - Termo de referência e estado da arte

Autores: Silvana Rossetto, Bruno Silvestre, Noemi Rodriguez, Adriano Branco, Raul Gabrich, Denis Aoki, Vinicius Lima, Douglas Santana

Data: 5 de fevereiro de 2014

Revisado: 11 de fevereiro de 2014

1. Descrição do projeto

O objetivo principal deste projeto é propor e avaliar uma arquitetura básica para a construção de ambientes de experimentação físicos (testbeds) que permitam experimentar aplicações de sensoreamento nas quais parte da aplicação executa em plataformas de nós sensores (em uma mesma sub-rede ou em sub-redes distintas) e outra parte executa em nós remotos em outras sub-redes sem fio ou conectados via Internet.

Com base no objetivo descrito acima, as seguintes metas deverão ser atingidas:

1. Disponibilização de um ambiente distribuído de experimentação remota de aplicações de sensoreamento para grupos de pesquisa no Brasil.
2. Proposta de um modelo de referência para construção de *testbeds* para redes de sensores com finalidades específicas, incluindo ferramentas para auxiliar na gerência e uso do *testbed*.
3. Implementação de soluções de controle de acesso federado para *testbeds* físicos locais, permitindo que grupos de pesquisas no Brasil disponibilizem seus recursos de hardware para acesso por outros grupos.
4. Implementação de soluções de software que permitam criar canais de comunicação entre nós em uma rede de sensores e outros dispositivos conectados a Internet usando protocolos padronizados.
5. Avaliação da arquitetura de *testbed* proposta por meio de aplicações finais e de algoritmos distribuídos específicos para aplicações de sensoreamento.

2. Cenário atual

Neste documento, apresentamos um relato do cenário atual de testbeds para redes de sensores sem fio e dos padrões de comunicação e implementações correspondentes para interconexão dessas redes com outras redes conectadas à internet.

2.1 Situação atual de testbeds para redes de sensores sem fio

A Tabela 1 mostra uma relação de testbeds para redes de sensores sem fio encontrados na literatura, com características em comum com a proposta do nosso trabalho. Priorizamos os testbeds para ambiente interno e que utilizam dispositivos (*motes*) com poucos recursos computacionais. Tomamos como referência para esse levantamento o relatório anterior feito pelo nosso grupo entitulado "*RT2-AMB Hardware do ambiente de experimentação".*

De forma geral, os motes utilizados nos testbeds analisados são equivalentes ao padrão de motes de Berkeley que tem como exemplo o mote MicaZ. A maioria dos testbeds encontrados é de origem europeia e americana. Com relação aos testbeds americanos, todos atuam de maneira independente, i.e, não estão disponíveis em uma federação. Nos testbeds europeus, por outro lado, apenas o *TWIST*, *SignetLab* e *SensLab* atuam de maneira

independente. Todos os demais encontram-se integrados em um grande portal denominado *Wisebed*.

O *Wisebed* é o portal de entrada principal para o acesso a testbeds de nove grandes institutos de tecnologia europeus. Entretanto, a forma de acesso a esses testbeds varia de instituição para instituição, podendo em alguns casos ser utilizado o próprio login do Wisebed (caso do *UZL*), em outros algum tipo de cadastro estabelecido pela própria universidade, ou autenticação federada. O Wisebed oferece algumas ferramentas de suporte para a execução de experimentos, incluindo a possibilidade de criação de scripts com plano de execução e frameworks para visualização dos experimentos (ex., SpyGlass).

Com relação aos testbeds asiáticos, encontramos apenas informações contidas em artigos.

Nome do Testbed	**País/Instituição**	**Qtd nós**	**Acesso**	**Estado do portal**
MoteLab	EUA / Harvard University	184	Restrito	Ativo
TU Delft	Países Baixos / TU Delft	140	Indisponível	Página Desativada
WSNTB	Taiwan / National Tsing Hua University	34	-	Artigo
TWIST	Alemanha / TU Berlin	204	Público*	Ativo
Trio testbed	EUA / UC Berkeley	564	-	Artigo
TutorNet	EUA / University of Southern California	104	Indisponível	Página Incompleta
IntelSensorNet	EUA / Intel Research laboratory in Berkeley, CA	148	Indisponível	Página Desativada
Kansei	EUA/ The Ohio State University	473	Publico*	Ativo
Emulab	EUA/ University of Utah	31	Publico*	Ativo
TUBS	Alemanha / Braunschweig University of Technolog	30	Restrito	Ativo
TUD	Países Baixos / TU Delft	140	Indisponível	Página desativada
UBERN	Suiça / University of Bern	47	Publico**	Ativo
UNIGE	Suiça / University of Geneva	28	Publico**	Ativo
FUB	Alemanha / Freie Universitat Berlin	110	Indisponível	Página Desativada
UNLAC	Inglaterra / Lancaster University (ULANC)	16	Restrito	Ativo
UZL	Alemanha / University of Lubeck	162	Publico	Ativo
UPC	Espanha / Universitat Politecnica de Catalunya	10	Indisponível	Página Desativada
TACTI	Grécia / Research Academic Computer Technology Institute	154	Restrito	Ativo
SignetLab	Itália / University of Padova	48	Restrito	Ativo
SensLab	França / Grenoble, Strasbourg, Lille e Bretagne	1024	Restrito	Ativo
Sensei-UU	Suécia / Uppsala University	Variável ***	-	Artigo
Re-Mote	Dinamarca/Kobenhavns Universitet Reino Unido/Cork Institute of Technology	-	Indisponível	Ativo

NetEye	EUA /Wayne State University	130	Publico*	Ativo
w-iLab	Bélgica/IBBT Ghent University Belgium	200	Restrito	Não encontrado
IRIS*	Irlanda/CTVR,Trinity College Dublin	8****	Publico*	Não encontrado
ISM/TVWS	Slovenia/Municipality of Miren–Kostanjevica, Jožef Stefan Institute	20		Não encontrado
iSense	India/School of Mobile Computing and Communication, Jadavpur University	-	-	Não encontrado
PARED	China/School of Instrumentation Science and Opto-Electronics Engineering Beihang University	50	-	Artigo
HINT	China/Institute of Software, Chinese Academy of Sciences; Canadá/School of Computing Science, Simon Fraser University	20 (Teste Apresentado)	-	Artigo
EASITEST	China/Chinese Academy of Science	-	-	Artigo
SenseNet	Grécia/Athens Information Technology Peania,	8 (Teste apresentado)	-	Artigo
MSRLab6*****	China/Beijing Jiaotong University	20	-	Artigo
IBM Wireless Sensor Networking Testbed	Suíça/IBM Zurich Research Laboratory	-	-	Artigo
WSNLab*****	Alemanha/University of Bonn	Variável ***	-	Artigo
Holistic IPv6 Test-Bed*****	Grécia/Computer Technology Institute & Press \Diophantus" (CTI) and University of Patras	35	-	Artigo

Tabela 1: Relação de testbeds para redes de sensores sem fio encontrados na literatura.

*Acesso é público, mas necessita de autorização dos administradores
**Acesso via VHO com problemas
***Suporta nós móveis, artigo não especifica a quantidade minima ou máxima de nós
****Os nós são constituídos de computadores
*****Utiliza ipv6

2.1.1 Testbeds europeus

I) **Wisebed (Ativos)** - http://wisebed.eu/site/conduct-experiments/testbeds/

- TU Delft Países Baixos / TU Delft
- TUBS Alemanha / Braunschweig University of Technology
- TUD Países Baixos / TU Delft
- UBERN Suiça / University of Bern
- UNIGE Suiça / University of Geneva
- FUB Alemanha / Freie Universit¨at Berlin
- UNLAC Inglaterra / Lancaster University (ULANC)
- UZL Alemanha / University of Lubeck
- UPC Espanha / Universitat Politecnica de Catalunya
- TACTI Grécia / Research Academic Computer TechnologyInstitute

II) Independentes

- TWIST Alemanha / TU Berlim (ativo- http://www.twist.tu-berlin.de/wiki)
- SignetLab Itália / University of Padova (descrição - http://telecom.dei.unipd.it/labs/read/3/)
- SensLab França / Grenoble, Strasbourg, Lille e Bretagne(ativo - http://www.senslab.info/)

Em artigo apenas: Sensei-UU Suécia / Uppsala University

2.1.2 Testbeds americanos

I) Independentes

- MoteLab USA / Harvard University(ativo - http://motelab.eecs.harvard.edu/)
- IntelSensorNet USA / Intel Research laboratory in Berkeley,CA (site fora do ar)
- Kansei USA / The Ohio State University(ativo - http://kansei.cse.ohio-state.edu/KanseiGenie/)
- Emulab USA / University of Utah(ativo - http://www.emulab.net/)
- Urban Test Bed(descrição - http://www.casa.umass.edu/main/research/urbantestbed/)
- TutorNet USA / University of Southern California(descrição - http://enl.usc.edu/projects/tutornet/)

Em artigos apenas: Trio testbed USA / UC Berkeley.

2.1.3 Testbeds asiáticos

I) Em artigos apenas

- WSNTB Taiwan / National Tsing Hua University
- MSRLab6 China / Beijing Jiaotong University
- EASITEST China / Chinese Academy of Science
- HINT China/Institute of Software, Chinese Academy of Sciences; Canadá/School of Computing Science, Simon Fraser University
- PARED China/School of Instrumentation Science and Opto-Electronics Engineering Beihang University
- iSense India/School of Mobile Computing and Communication, Jadavpur University

2.1.4 Avaliação

A arquitetura de testbed proposta neste projeto se assemelha com as arquiteturas dos testbeds elencados. Podemos notar que a quantidade de nós disponíveis varia bastante de um testbed para o outro. No nosso caso teremos até 45 nós de dois modelos distintos (Micaz e TelosB). Com relação às ferramentas de suporte e gerência dos testbeds, tomaremos o exemplo do Wisebed como modelo de referência, particularmente com respeito à alternativa de autenticação federada. Não encontramos nenhum testbed com ênfase em permitir experimentos envolvendo nós em uma rede de sensores e outros dispositivos conectados a Internet usando protocolos padronizados.

2.2 Situação atual dos padrões 6LoWPAN e suas implementações

Nesta seção, apresentamos um relato sobre os esforços de implementação da pilha de protocolos Internet nos dispositivos que formam as redes LoWPAN (*Low power Wireless Personal Area Networks*), nas quais as redes de sensores sem fio de enquadram.

2.2.1 O padrão 6LoWPAN

6LoWPAN é um acrônimo para "*IPv6 over Low power Wireless Personal Area Networks*" (Kushalnagar, N., Montenegro, G., and Schumacher, C. (2007)). Trata-se, inicialmente, de um grupo de trabalho do IETF responsável pela definição de normas (como formato de quadros) que permitam a implementação do protocolo IPv6 sobre redes IEEE802.15.4 (padrão da camada de enlace usado nas redes LoWPAN). Atualmente, a implementação de redes LoWPAN é efetuada por um conjunto de tecnologias proprietárias, o que dificulta a interoperabilidade entre diferentes redes e a sua inserção em serviços baseados na Internet. O esforço de integração do protocolo IP em redes LoWPAN permite superar este problema e introduz um conjunto de vantagens:

- a natureza ubíqua das redes IP permite a utilização de infra-estruturas de rede já existentes;
- a tecnologia IP é bem conhecida e já se encontra devidamente testada;
- o protocolo IP é definido em especificações do IETF, as quais são disponibilizadas publicamente;
- já existem um conjunto de ferramentas disponíveis para diagnóstico e gestão de redes IP;
- dispositivos com conectividade IP podem ser ligados a uma rede IP sem necessidade de um gateway ou proxy.

No entanto, os dispositivos que tipicamente formam as redes IEEE802.15.4 possuem várias limitações que dificultam a implementação do protocolo IP, entre elas:

- o alcance de transmissão dos nós é de apenas algumas dezenas de metros;
- a taxa de transmissão máxima é de 250 kbps;
- os recursos de memória e fonte de energia são limitados;
- e o padrão IEEE802.15.4 limita o tamanho dos quadros de enlace em 127 bytes.

Por isso, a adoção do protocolo IP em redes IEEE802.15.4 traz uma série de desafios, comentados em Shelby, Z., Hartke, K., and Bormann, C. (2013).

Entre os objetivos do grupo 6LoWPAN está o de definir mecanismos para acomodar pacotes IPv6 (cujo MTU mínimo é de 1280 bytes) em quadros IEEE802.15.4 (com MTU de

apenas 127 bytes). Para isso, uma camada de adaptação é acrescentada à pilha IP, entre a camada de rede e a camada de enlace, para permitir a fragmentação e desfragmentação de pacotes IPv6 em quadros IEEE802.15.4, e efetuar a compressão do cabeçalho IPv6 (RFC 4919). A Figura 1 mostra a pilha de protocolos do modelo de referência 6LoWPAN.

As recomendações 6LoWPAN, definidas na RFC4919 e na RCF4944, indicam ainda os requisitos para efetuar o encaminhamento de pacotes em redes de malha, sugere algumas recomendações de segurança e descreve alterações no protocolo de descoberta de vizinhos (*Neighbor Discovery Protocol* - NDP) de forma a otimizá-lo para redes LoWPAN. O protocolo NDP tem as seguintes funções: auto-configuração de endereços, resolução de endereços, detecção de duplicação de endereços e descoberta de roteadores (RFC4861).

2.2.2 Desafios para implementação da pilha IPv6 em redes LoWPAN

Com base nas características das redes LoWPAN é possível elencar desafios e problemas a serem tratados pelo protocolo da camada de adaptação. Alguns desses desafios são descritos brevemente a seguir.

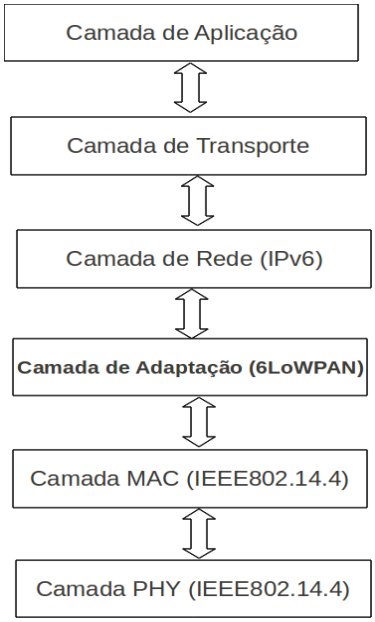

Figura 1: Pilha de camadas no modelo de referência 6LoWPAN.

Conectividade IP

Como as redes LoWPAN são dinâmicas, a conectividade IP deve ser feita através da atribuição dinâmica de endereços. Além disso, a rede pode ter um número potencialmente grande de dispositivos a ela conectados (esse é o caso, por exemplo, das redes de sensores sem fio). Portanto, deve existir suporte adequado para manipular um número considerável de endereços de rede/objetos endereçáveis. Ambas as características já possuem soluções provindas do IPv6.

Adequação a diferentes topologias

O padrão deve permitir o seu uso em topologias diversas, incluindo topologia em malha e estrela. A topologia em malha deve permitir roteamento multi-saltos, o que exige que os dispositivos tenham a capacidade de reencaminhamento de pacotes. Os seguintes requisitos se apresentam para os protocolos de roteamento:

- devido ao limite do tamanho dos quadros, o protocolo deve evitar perdas nos pacotes de dados, independente do número de saltos;
- o protocolo deve minimizar as trocas de mensagens, independentemente da topologia empregada;
- o protocolo deve ser simples (baixa complexidade computacional e uso da memória) para que seja satisfeito o requisito de baixo consumo de energia, característico das redes LoWPAN.

A topologia em estrela também deve possuir um conjunto de dispositivos responsáveis por fazer o encaminhamento das mensagens. Se for necessário ter outros tipos de interface de rede para obter isso, o desafio será integrar essas redes.

Limitação do tamanho do pacote

O tamanho mínimo dos pacotes IPv6 é de 1280 bytes, o que faz com que um pacote IPv6 não caiba em um quadro IEEE802.15.4, que possui somente 127 bytes de tamanho máximo. Como o padrão IEEE802.15.4 requer 25 bytes de cabeçalho, restam apenas 102 bytes de payload para a camada superior. Dessa forma, faz-se necessário empregar mecanismos de compactação de cabeçalho, e de fragmentação e remontagem de pacotes.

Descoberta de Serviço

As redes LoWPAN devem ter um protocolo de descoberta de serviço simples para que seja possível controlar e manter os serviços fornecidos pelos dispositivos.

Gerenciamento da rede

Um dos objetivos de utilizar IPv6 é possibilitar a reutilização de diversos protocolos conhecidos, como uma adaptação do protocolo SMNPv3 para que fique apropriado às características da rede.

Demandas para as camadas superiores

Como a compressão de pacotes é um aspecto importante para o bom funcionamento da rede, o formato e quantidade de dados gerados pela camada de aplicação terão impacto direto nas camadas inferiores. Algumas abordagens de transmissão de dados (ex., XML, SOAP) se tornam inviáveis para redes LoWPAN, o que torna necessária a criação ou modificações de protocolos de aplicação com o objetivo de adaptá-los a esse tipo de rede.

Considerações de segurança

As ameaças de segurança devem ser claramente identificadas, entendidas e documentadas. A especificação IEEE802.15.4 já define uma proteção na camada de enlace, baseada no algoritmo de criptografia AES2. No entanto, não detalha alguns aspectos, como gerenciamento de chaves.

Modo de endereçamento

O padrão IEEE802.15.4 permite endereçamento de 64 bits e endereçamento curto de 16 bits para regiões da rede. Além disso, possui endereçamento broadcast e unicast, porém não admite endereçamento multicast. O IPv6, por sua vez, possui endereçamento multicast e não admite endereçamento broadcast. A recomendação é então fazer o compartilhamento dos recursos de ambas as redes, fazendo com que uma mensagem IPv6 de multicast mapeie uma mensagem de broadcast no enlace IEEE802.15.4.

Compressão de cabeçalhos

A RFC6282 (Hui, J. and Thubert, P. (2011)) define uma estratégia de compressão de cabeçalho para a pilha IPv6 a fim de reduzir o tamanho do cabeçalho IPv6 e dar mais espaço para os dados. O cabeçalho IPv6 comprimido é chamado LOWPAN_IPHC e assume que os seguintes campos do cabeçalho IPv6 serão de uso comum nas comunicações em redes 6LoWPAN:

- *Version* é sempre 6;
- *Traffic Class* e *Flow Label* são ambos iguais a 0;
- *Payload Length* pode ser inferido das camadas inferiores (cabeçalho de fragmentação 6LoWPAN ou cabeçalho IEEE 802.15.4);
- *Hop Limit* pode receber um valor padrão;
- os endereços designados para interfaces 6LoWPAN podem ser formados usando o prefixo de enlace local ou um pequeno conjunto de prefixos de roteamento designados para 6LoWPAN.

A compressão, no melhor caso, pode reduzir o cabeçalho IPv6 de 40 para 2 bytes. A Figura 2 mostra o exemplo de um cabeçalho compactado (Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T., and Roedig, U. (2011)).

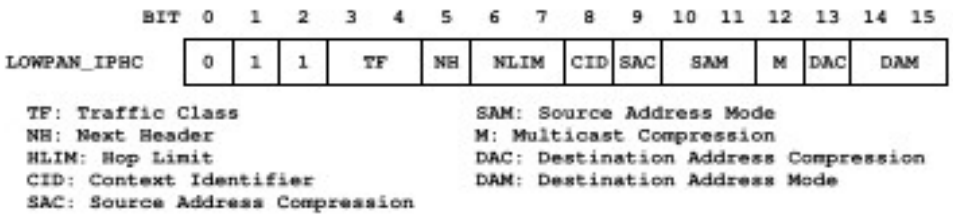

Figura 2: Exemplo de um cabeçalho IPv6 compactado.

2.2.3 Redes LoWPAN como redes de acesso à Internet

A proposta de integração das redes LoWPAN com a Internet consiste em fazer com que as redes LoWPAN operem como redes de acesso a Internet, conectando-se a outras redes IP através de um ou mais roteadores de borda que redirecionam datagramas IP através de diferentes mídias. A Figura 3 ilustra essa proposta de organização.

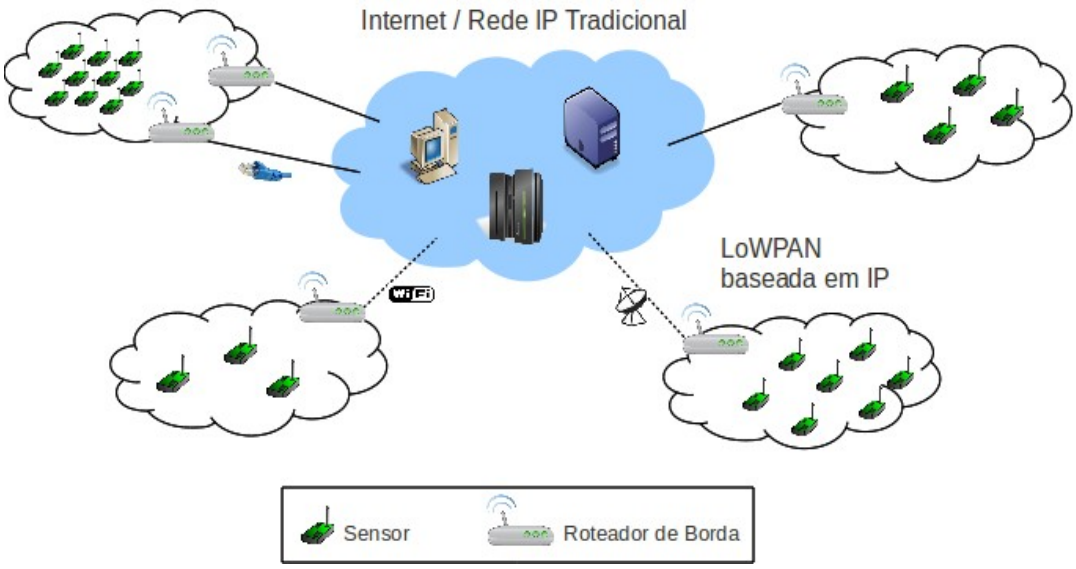

Figura 3: Proposta de extensão da arquitetura Internet para incluir redes LoWPAN.

2.2.4 Experiências de implementação da pilha TCP/IP em dispositvos de baixo consumo

Avaliamos algumas experiências recentes de implementação da pilha TCP/IP em plataformas de sensores. Começamos pelo trabalho pioneiro do grupo do prof. Dunkels, voltado para o sistema operacional Contiki. Os resultados desse trabalho serviram de motivação para a criação pela IETF do grupo 6LoWPAN. Em seguida avaliamos algumas implementações da arquitetura 6LoWPAN, em particular aquelas que fazem uso do sistema operacional TinyOS (um dos sistemas operacionais mais usados na pesquisa com RSSFs).

uIP e lwIP no sistema Contiki

Entre os primeiros esforços para implementar a pilha de protocolos IP em plataformas de sensores está o trabalho realizado pelo grupo que desenvolveu e mantém o sistema operacional Contiki (Dunkels, A., Gronvall, B., and Voigt, T. (2004)) (um sistema operacional para dispositivos com restrições de recursos computacionais, como é o caso dos dispositivos que formam as RSSF). O Contiki foi desenvolvido em C e possui versões para vários microcontroladores, entre eles o MSP430 e microcontroladores da família Atmel-AVR. A implementação atual segue as recomendações da RFC-4944 e a da RFC-6282.

O artigo Dunkels, A. (2003) discute duas implementações da pilha TCP/IP: **lwIP** (*lightweight IP*) e **uIP** (*microIP*). O lwIP é uma implementação simplificada que permite várias interfaces de rede e provê os protocolos IPv4, ICMP, UDP e TCP. O uIP provê o conjunto mínimo de funcionalidades necessárias para uma pilha TCP/IP e oferece suporte a apenas uma interface de rede. Como o objetivo do trabalho era avaliar o tamanho do código das implementações, mostrando que a implementação da pilha IP em dispositivos de recursos limitados é possível, aspectos como configuração de endereço, segurança e consumo de energia não foram abordados. Pelo mesmo motivo, o trabalho se concentra apenas nos protocolos TCP e IP, abstraindo tanto as camadas superiores, de responsabilidade da aplicação, quanto as camadas inferiores, geralmente implementadas em hardware ou firmware.

Nas implementações lwIP e uIP, alguns requisitos das especificações dos protocolos da pilha TCP/IP foram desconsiderados para reduzir o tamanho do código implementado. Esses requisitos foram escolhidos de tal forma que não impedissem a comunicação computador-para-computador. Apesar de algumas restrições terem sido colocadas com relação ao tamanho de buffer, e a retransmissão de pacotes, elas limitam o desempenho da pilha, mas não impedem o seu funcionamento.

Memória é um dos recursos mais escassos nas plataformas de sensores. Para o recebimento de pacotes, lwIP permite a alocação dinâmica de buffers para manter informações sobre as conexões e pacotes recebidos. A pilha uIP não permite essa alocação dinâmica. Ela trabalha com um único buffer global capaz de manter um pacote, e mantém uma tabela de tamanho fixo para armazenar o estado da conexão. Dessa forma, a aplicação deve tratar o pacote recebido imediatamente, ou realizar um cópia dele, liberando espaço para o recebimento de um novo pacote.

Para o envio de pacotes, lwIP também trabalha com a alocação dinâmica de buffers de envio, que só são liberados depois que a transmissão for bem sucedida (mantendo os buffers em caso de retransmissão). Um mecanismo adicional permite que a aplicação informe se os dados a serem enviados são voláteis ou não. Com base nessa informação, é possível avaliar a necessidade de copiar os dados da aplicação para um buffer de saída (no caso de dados voláteis) ou somente apontar para os dados (no caso de dados não voláteis). Dessa forma é possível reduzir o uso de memória no sistema.

A uIP utiliza o mesmo buffer global de recebimento para o envio. O buffer é usado para manter o cabeçalho que é calculado para o pacote a ser enviado. A aplicação passa os dados para a pilha uIP, esta produz o cabeçalho e o envia pela rede, juntamente com os dados fornecidos pela aplicação (o pacote não é armazenado internamente).

Em termos de interface de programação, diferentemente da API geralmente usada nos sistemas operacionais mais populares, que seguem a API de socket do BSD, as pilhas implementadas empregam um modelo orientado a eventos. A API do BSD usa a ideia de suspender a processo que está enviando ou recebendo mensagens, o que leva a uma troca de contexto. Esse procedimento mais elaborado não é uma boa opção para as plataformas de sensores pois demanda muitos recursos para realizar o controle de troca de contexto. A abordagem de eventos quebra o processo de transmissão em estágios, facilitando a gerência da tarefa de envio e recebimento de mensagens por parte da pilha.

Como o protocolo IP permite a fragmentação dos pacotes a serem enviados de acordo com o tamanho dos quadros da camada inferior, ambas implementações reservam um único buffer adicional para a remontagem do pacote original.

Tanto lwIP e uIP implementam o protocolo TCP, mas algumas adaptações e simplificações foram necessárias. Um exemplo é o mecanismo de janela deslizante. Enquanto que na pilha lwIP, o mecanismo está disponível normalmente, na pilha uIP ele não é implementado pois requer um espaço adicional para armazenar os pacotes já enviados e aguardando confirmação de recebimento. Caso ocorra um erro, a aplicação utilizando a uIP deve gerar novamente o mesmo conjunto de dados e passá-los para a pilha efetuar a retransmissão. Temos, nesse caso, um exemplo de como a adaptação da pilha IP para dispositivos de baixa potência pode afetar o modo como a aplicação deve ser desenvolvida. O fato da uIP não oferecer janela deslizante e retransmissão de pacotes afeta os mecanismos de

controle de fluxo e congestionamento. Como ela permite que somente um pacote esteja na rede por vez, esses controles são simplificados ou até inexistentes.

BLIP (TinyOS)

Entre as implementações existentes das recomendações do grupo 6LoWPAN para o sistema operacional TinyOS (Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. (2000), uma das mais importantes é a Blip (Berkeley IP Information).

A implementação do Blip disponibilizada atualmente está na versão 2.0, lançada em maio de 2010 para o TinyOS versão 2.1.1, e apresenta conformidade total com a RFC4944. Blip 2.0 usa compressão do cabeçalho 6lowpan/HC01 e inclui a implementação do *IPv6 Neighbor Discovery Protocol*, além de outras características adicionais, como seleção de rota default e roteamento ponto-a-ponto. O Blip foi implementado para as plataformas MicaZ, TelosB e Epic, e o seu código atual foi testado em redes de até 75 nós, para análise de estabilidade e desempenho.

O Blip oferece suporte a diversas funcionalidades do IPv6, dentre as mais importantes, a descoberta de vizinhos e o roteamento ponto-a-ponto. Além disso, o protocolo tem suporte ao ICMP, UDP e TCP, embora a implementação do TCP ainda esteja em caráter experimental e possa não oferecer o desempenho e a confiabilidade usuais do TCP. A conexão entre a RSSF e a Internet é feita através de um driver de tunelamento para Linux. Desse modo, um computador executando o sistema operacional Linux opera como um roteador de borda, repassando os pacotes para a Internet. O driver também implementa o protocolo de descobrimento de vizinhos do IPv6, permitindo que os nós da rede encontrem o roteador de borda.

O endereçamento, a autoconfiguração sem estados e a compressão de cabeçalho do Blip é feito em conformidade com a RFC4944. Como a rede de sensores utilizando Blip pode ser vista como qualquer outra rede IP, muitos programas conhecidos para computadores também podem ser utilizados, dentre eles: *ping6*, *tracrt6* e *nc6*.

A- Componentes desenvolvidos pelo Blip

Por se tratar de uma implementação para o sistema operacional TinyOS, o Blip é desenvolvido utilizando nesC (Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E., and Culler, D. (2003)), uma extensão da linguagem C baseada em componentes. Assim, o Blip é facilmente separado em componentes. A Figura 4 mostra os componentes principais do código Blip. Os componentes da camada de rede são:

- *IPRouting*: implementa a funcionalidade de descoberta de vizinhos do IPv6 (*Neighbor Discover Protocol*), constrói a tabela de roteamento e define o próximo nó intermediário na rota de um pacote;
- *IPDispatch*: trata as requisições de encaminhamento de pacotes recebidas da camada de transporte usando os serviços da camada de enlace, realiza a fragmentação de pacotes e a validação das transmissões;
- *IPAddress*: obtem os endereços IPv6 e IEEE802.15.4 no nó.

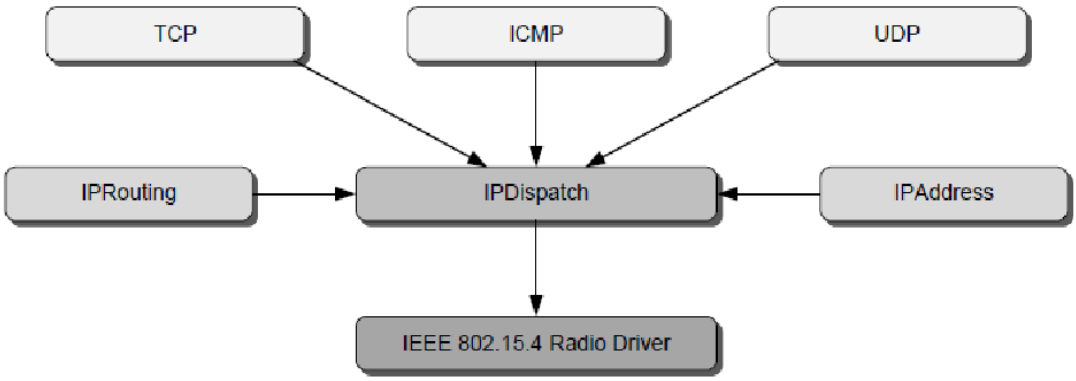

Figura 4: Componentes principais do Blip.

O componente ICMP (*Internet Control Message Protocol*) faz requisições de envio ao roteador. O componente UDP é responsável pelos pacotes UDP e provê funcionalidades para facilitar o recebimento e envio de pacotes UDP. No componente TCP são implementadas funções de envio e recebimento de pacotes que visam garantantir entrega confiável (esse componente funciona apenas de forma experimental e não implementa todas as definições do protocolo TCP).

Para funcionar, o Blip requer que uma estação base conecte a RSSF a um computador. Tanto o computador quanto a estação base implementam a funcionalidade de roteador de borda. O código na estação base não precisa da camada IP e nem da camada de transporte, e acessa o driver do rádio diretamente. Assim, os pacotes são diretamente encaminhados para o computador e vice-versa.

B- Problemas na implementação atual

Além do já citado problema com a implementação do TCP, existe ainda no Blip dois outros problemas na sua implementação. O primeiro deles é que a fragmentação de pacotes apresenta diversos erros, particularmente quando é utilizado o redirecionamento multisaltos. A única solução conhecida até o momento é utilizar mensagens pequenas, o que elimina a necessidade de usar a fragmentação. O outro problema é relacionado aos buffers de mensagens. Esses buffers acabam consumindo muita memória, e ao reduzir o tamanho máximo do buffer, mensagens acabam sendo descartadas. Esse problema é mais evidente em plataformas que possuem apenas 4Kbytes de memória RAM. Em nós com 8Kbytes, que suportam janelas de buffer maiores, o problema não é tão comum.

2.2.5 Algoritmo de roteamento RPL

As redes LoWPAN são definidas como redes nas quais os roteadores tipicamente operam com severas restrições em termos de capacidade de processamento, armazenamento e fonte de energia. As interconexões de comunicação, por sua vez, são caracterizadas por elevadas taxas de perdas de pacotes, baixas taxas de transmissão e alta instabilidade. Para lidar com esses desafios particulares e minimizar o *overhead* de controle, a IETF propôs um protocolo de roteamento IPv6 específico, chamado RPL (RFC6550) (Winter, T. and et.al. (2012)).

O RPL é um protocolo do tipo *distance vector* que suporta três categorias de padrões de tráfego. Na primeira, *multipoint-to-point*, os nós periodicamente enviam mensagens para um ponto de coleta específico, por exemplo um nó *sink*. Na segunda, *point-to-multipoint*, o tráfego originado em um nó *sink* tem como destino dispositivos específicos dentro da rede. Por último, a comunicação *point-to-point* também é suportada.

O RPL introduz uma série de conceitos que o torna bastante flexível, embora relativamente complexo. A formação da topologia da rede é baseada no conceito de Grafos Acíclicos Dirigidos (*Directed Acyclic Graph -- DAG*) --- uma estrutura em forma de árvore que define rotas default entre nós da rede. Para aumentar a confiabilidade, o RPL adota o conceito de "diversidade espacial". Assim, diferentemente de uma árvore tradicional, onde um nó está associado a um único nó pai, em um grafo RPL um nó pode estar associado a vários nós pais, criando caminhos alternativos em direção a um destino.

O protocolo organiza os nós como um conjunto de *Destination Oriented DAGs* (DODAGs). Nos DODAGs, os nós mais populares, como o nó *sink* e os nós gateways, são os nós raízes dos DAGs. Uma instância do protocolo RPL (*RPL Instance*), identificada univocamente por um *RPLInstanceID*, pode conter múltiplos DODAGs, cada um deles identificado por um único DODAGID. Cabe observar que uma rede LoWPAN pode ter várias instâncias RPL executando concorrentemente.

O RPL suporta aplicações com diferentes requisitos por meio da definição de Funções Objetivo (*Objective Functions - OFs*). Basicamente, uma OF especifica como o RPL deve selecionar seus pais e possíveis sucessores, ou seja, como selecionar caminhos no DODAG. As OFs definem de que maneira métricas de roteamento e funções relacionadas são empregadas pelos nós para computarem o seu *Rank* dentro de um DODAG. Em última instância, as OFs restringem ou otimizam as rotas selecionadas. Para garantir interoperabilidade de comunicação entre diferentes aplicações, o RPL provê uma função objetivo básica denominada **OF0** (Thubert, P. (2011)), a qual seleciona rotas com base no número de saltos até o nó raiz do DODAG. Observa-se ainda que é possível definir OFs por aplicação (ex: prover rotas sensíveis à latência para aplicações de tempo real).

Para construir e manter o DODAG, o protocolo RPL define mensagens ICMPv6 denominadas *DODAG Information Object* - DIO para a descoberta de vizinhos e o estabelecimento de rotas. Na formação da topologia, cada nó raiz constrói um pacote DIO e o envia para todos os filhos. Qualquer filho que decide se juntar ao DAG repassa o DIO adiante, para os seus próprios filhos. O DIO contém um valor do Rank do nó, que é incrementado quando o filho se junta ao DAG. Além disso, os nós podem armazenar um conjunto de pais e irmãos candidatos, que podem ser usados se o pai preferencial está incapacitado de rotear tráfego. O DIO também é usado para indicar a OF. Por fim, a propagação de rotas é implementada com o algoritmo *Trickle* (Levis, P., Clausen, T., Hui, J., Gnawali, O., and Ko, J. (2011)), o qual escalona o envio de mensagens DIO visando sempre minimizar a quantidade de DIOs transmitidos e garantir um tempo de convergência baixo para a rede.

Como redes LoWPAN são bastante dinâmicas, o RPL também fornece facilidades para incorporar métricas de natureza dinâmica, tais como (ETX - *Estimated number of Transmissions*} (Couto, D., Aguayo, D., Bicket, J., and Morris, R. (2005)) (facilita a descoberta de caminhos de maior vazão e minimiza o número total de transmissões de um pacote até o seu destino). Além disso, se um nó detecta a inexistência de uma rota em direção à raiz, um mecanismo de "reparo local" é acionado para encontrar uma rota alternativa. Quando uma

sequência de reparos locais levarem a uma topologia da árvore ineficiente, um mecanismo de "reparo global" pode ser usado (Korte, K., Sehgal, A., and Schönwälder, J. (2012)).

Tanto o Contiki quanto o TinyOS possuem implementações para o protocolo RPL integradas em sua base e prontas para uso (ContikiRPL e TinyRPL, respectivamente).

2.2.6 Protocolos na Camada de Aplicação

Em nosso levantamento mais recente sobre o estado da arte de IPv6 para RSSF, notamos que os padrões que descrevem a pilha 6LoWPAN não sofreram atualizações, com isso suas principais implementações se mantiveram também de certa forma estável. Até a escrita deste relatório, o TinyOS não havia lançado nova versão e a lista de discussão do BLIP estava relativamente pouco acessada. O Contiki lançou a versão 2.7, focando mais na estabilidade do código, mas não apresentou nenhuma alteração significante relacionado ao 6LoWPAN.

Por outro lado, estamos vendo a migração e o aparecimento de protocolos na camada mais alta da pilha, i.e., na camada de aplicação. Isso significa que a pilha está sendo considerada estável e que há um esforço em levar as funcionalidades que as aplicações baseadas em IP necessitam para as RSSFs. Com a construção de um ambiente compatível com o que temos hoje na Internet, pode ser cada vez mais fácil interconectar as aplicações com as RSSFs. A seguir, descrevemos rapidamente alguns exemplos de migração de protocolo.

IPSec

Aplicações de RSSFs necessitam de segurança na comunicação (por exemplo, medidores inteligentes, monitoramento de pacientes). O protocolo IPv6 já possui suporte à IPSec por definição, dessa forma, possuir IPSec integrado com 6LoWPAN é desejável.

No artigo (Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T., and Roedig, U. (2011)), os autores demonstram a forma de integrar IPSec com a proposta atual de 6LoWPAN. Eles também apresentaram uma proposta de RFC que está em avaliação. Os testes foram feitos utilizando a pilha uIP do Contiki. Essa proposta consiste em adaptar os protocolos *Authentication Header* (AH) e *Encapsulating Security Payload* (ESP) do IPSec, seguindo a ideia de compactação de cabeçalho. O protocolo *AH* apresentou um overhead de 16 bytes (original: 24 bytes), ESP de 12 bytes (original: 16 bytes) e AH em conjunto com ESP, de 24 bytes (original: 30 bytes).

O padrão IEEE802.15.4 define o uso de segurança salto-a-salto. Assim, alguns chips de rádio já possuem suporte à criptofrafia embutido. Eles pondem ser usados para a melhor desempenho do IPSec (dado que geralmente a cryptografia via software é muito mais lenta).

SNMP

O protocolo SNMP (*Simple Network Management Protocol*) permite aos administradores de redes gerenciarem os equipamentos de rede, principalmente para finalidade de monitoramento. O SNMP funciona basicamente sobre dois elementos: um supervisor e agentes. O supervisor é o dispositivo utilizado pelo administrador de rede para executar pedidos de gestão. Os agentes são aplicações de gestão de rede que residem

em um dispositivo periférico e encarrega-se de transmitir os dados locais de gestão no formato SNMP.

A arquitetura SNMP e suas amplamente usadas aplicações NMS (Network Management Systems) são ideais para o gerenciamento 6LoWPAN. A arquitetura modular flexível dos frameworks SNMP é adequada para nós sensores com recursos limitados, porque apenas um conjunto desejado de recursos pode ser implementado independentemente e o framework é bem adequado para estender as funcionalidades de apoio ao 6LoWPAN.

Em (Choi, H., Kim, N., and Cha, H. (2009)), o esforço dos autores para implementar o protocolo SNMP sobre o 6LoWPAN concentrou-se em comprimir os header utilizados pelo SNMP e reduzir as mensagens de controle geradas. Além disso, visando reduzir o número de mensagens geradas pelo SNMP, uma das técnicas utilizadas foi a adoção do modo broadcast --- o padrão de comunicação do SNMP é o ponto-a-ponto. Em vez de gerar uma mensagem para cada nó na rede, periodicamente o NMS injeta uma mensagem broadcas Get Request na própria rede. Cada nó irá imediatamente responder ao NMS com a mensagem Response. Dessa forma uma única mensagem pode iniciar várias respostas.

DPWS

O DPWS (*Device Profile for Web Services*) foi basicamente criado para que RSSFs possam oferecer serviços Web (Samaras, I., Hassapis, G., and Gialelis, J. (2013)). Também visa prover uma forma das aplicações (por exemplo, monitoramento industrial e controle de sistemas) acessarem, utilizarem e recuperarem dados remotamente das RSSFs. A modificação proposta para o DPWS é chamada Tiny SOAWS (*Tiny SOA for Wireless Sensors*). Ela visa prover um serviço Web com a mesma semântica de um servidor DPWS e suprir dados para os clientes DPWS utilizando os padrões de interação DPWS. O DPWS utiliza um método de escalonamento para minimizar as colisões e perdas de pacotes, sendo que os parâmetros do método de escalonamento são definidos no padrão IEEE802.15.4. Os testes foram feitos em nós SunSPOT (os quais possuem capacidade de processamento relativamente grande se comparado com as plataformas MicaZ e TelosB, adotadas no nosso projeto).

CoAP

CoAP (*Constrained Application Protocol*) é um protocolo de transferência de dados para nós e redes limitados (voltado para o domínio da *Internet das Coisas) (*Shelby, Z., Hartke, K., and Bormann, C. (2013)). Parecido com o protocolo HTTP, o protocolo CoAP utiliza uma arquitetura REST, porém é consideravelmente menos complexo. Além disso, ele permite o desenvolvimento de *web services* até mesmo para dispositivos e redes mais limitados. Padronizado pela IETF e otimizado para aplicações de *Smart Energy* e automação residencial e predial, CoAP é uma das tecnologias chave para viabilizar Internet das Coisas.

No que se refere a sua adequação ao padrão 6LoWPAN, um elemento central do protocolo CoAP é sua complexidade reduzida. Ao invés de TCP, utiliza-se UDP e uma simples camada de mensagens para retransmissão de pacotes perdidos. Dentro dos pacotes UDP, CoAP utiliza um cabeçalho binário de quatro bytes, seguido por uma sequência de opções. Essa codificação compacta, mas facilmente analisável permite um tamanho total do cabeçalho de 10 a 20 bytes para uma requisição usual. No topo da camada

de mensagens, a especificação do protocolo CoAP define quatro métodos de requisição: GET, POST e DELETE. Além disso, códigos de retorno são padronizados similarmente aos códigos de retorno HTTP, porém são codificados em um único byte.

A arquitetura REST do protocolo CoAP permite que redes tradicionais utilizando HTTP se comuniquem com redes utilizando o protocolo CoAP através de proxies, intermediários que hora se comportam como clientes, hora como servidores.

O protocolo CoAP utiliza uma abordagem assíncrona para o envio de informações de servidores para clientes. Em uma requisição GET, um cliente pode indicar o seu interesse em novas atualizações de um recurso, especificando a “opção observar”. Se o servidor aceitar essa opção, o cliente se torna um observador do recurso e recebe mensagens de notificação assíncronas a cada vez que o recurso se altera.

mDNS

A integração de objetos inteligentes na infraestrutura de Internet atual exige um esquema padronizado para mecanismos de descoberta de objetos, respeitando o padrão IP da Internet, bem como a autoconfiguração ("chegar e operar") para lidar com o possível grande número de objetos enfatizados pela visão de Internet das Coisas.

O protocolo mDNS permite mapear nomes de domínios em endereços da Internet sem a ajuda de um servidor na rede local. Existem implementações deste protocolo para objetos inteligentes, uma delas é o *uBonjour* (desenvolvida sobre o Contiki) (Klauck, R. and Kirsche, M. (2013)). Através desta implementação, é possível realizar a descoberta e endereçamento de dispositivos e serviços disponíveis em ambiente de rede baseados em IP. Dessa forma, sistemas intermediários não são mais necessários para conectar objetos inteligentes com dispositivos computacionais comuns. Protocolos de aplicação podem registrar e anunciar sua disponibilidade como serviços na rede assim como descobrir outros dispositivos que utilizam o mesmo protocolo de aplicação.

Para adaptação ao padrão 6LoWPAN, foram definidas algumas regras de compressão. A razão para o pequeno tamanho do payload IP para dados da aplicação é que os cabeçalhos IP e UDP tomam uma grande parte do espaço de cada pacote enviado. Como o comprimento mínimo para todos os quatro registros de recursos DNS é 108 bytes, uma única mensagem de DNS não pode ser usada sem otimizações adicionais. Com as técnicas de compressão, foi possível chegar a um payload de 52 bytes, compatível com o padrão DNS.

Referências bibliográficas

Choi, H., Kim, N., and Cha, H. (2009). 6LoWPAN-SNMP: Simple network management protocol for 6LoWPAN. In 11th IEEE International Conference on High Performance Computing and Communications.

Couto, D., Aguayo, D., Bicket, J., and Morris, R. (2005). A high-throughput path metric for multi-hop wireless routing. Wireless Networks, 11(4):419–434.

Dunkels, A. (2003). Full TCP/IP for 8-bit architectures. In Proceedings of the 1st International Conference on Mobile Systems Applications and Services (MobiSys), pages 85–98. ACM Press.

Dunkels, A., Gronvall, B., and Voigt, T. (2004). Contiki — a lightweight and flexible operating system for tiny networked sensors. In Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, pages 455–462. IEEE.

Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E., and Culler, D. (2003). The nesC language: A holistic approach to networked embedded systems. Proceedings of Programming Language Design and Implementation (PLDI). http://doi.acm.org/10.1145/781131.781133.

Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., and Pister, K. (2000). System architecture directions for networked sensors. In 9th International Conference on Architectural Support for Programming Languages and Operating Systems, pages 93–104. ACM Press.

Hui, J. and Thubert, P. (2011). Compression format for IPv6 datagrams over IEEE 802.15.4-based networks. RFC6282.

Klauck, R. and Kirsche, M. (2013). Enhanced DNS message compression - optimizing mDNS/DNS-SD for the use in 6LoWPANs. In IEEE International Conference on Pervasive Computing and Communications Workshops.

Korte, K., Sehgal, A., and Schönwälder, J. (2012). A study of the RPL repair process using ContikiRPL. Dependable Networks and Services, pages 50–61.

Kushalnagar, N., Montenegro, G., and Schumacher, C. (2007). IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals. RFC4919.

Levis, P., Clausen, T., Hui, J., Gnawali, O., and Ko, J. (2011). The trickle algorithm. RFC6206.

Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T., and Roedig, U. (2011). Securing communication in 6lowpan with compressed ipsec. In Distributed Computing in Sensor Systems and Workshops (DCOSS), pages 1–8. IEEE.

Samaras, I., Hassapis, G., and Gialelis, J. (2013). A modified DPWS protocol stack for 6LoWPAN-based wireless sensor networks. IEEE Transactions on Industrial Informatics, 9(1).

Shelby, Z., Hartke, K., and Bormann, C. (2013). (CoAP). Internet-Draft. Constrained application protocol .

Thubert, P. (2011). RPL objective function zero. Draft-ietf-roll-of0-20. TinyOS-Blip (2012). docs.tinyos.net/tinywiki/index.php/blip_2.0.

Winter, T. and et.al. (2012). RPL: IPv6 routing protocol for low-power and lossy networks. RFC6550.