Acesso Remoto a Redes de Sensores Sem Fio Usando VPN

Matheus Nascimento[1], Bruno Silvestre[1], Silvana Rossetto[2]

[1]Instituto de Informática – UFG

{matheusnacimento,brunoos}@inf.ufg.br

[2]Departamento de Ciência da Computação – UFRJ

silvana@dcc.ufrj.br

Abstract. *In this work we present a solution based on VPN to connect private WSN (communicating in IPv6), with computers on the Internet (IPv4).*

Resumo. *Neste trabalho apresentamos uma solução baseado em VPN para oferecer conexão a RSSFs privadas (com comunicação IPv6) com computadores ligados na Internet (IPv4).*

1. Introdução

A crescente demanda pela construção dos chamados "espaços inteligentes" tem fomentado a investigação e o uso combinado de tecnologias de sensoreamento e comunicação sem fio acopladas a dispositivos de tamanho reduzido, com capacidade de processamento e armazenamento de dados, e fonte de energia independente [Estrin et al. 2002].

O desenvolvimento de aplicações baseadas nessas tecnologias requerem a construção de redes específicas — Redes de Sensores Sem Fio (RSSFs) — formadas por dezenas, centenas ou até milhares desses pequenos dispositivos [Yick et al. 2008]. A miniaturização e o enredamento dos sensores possibilitam diferentes ângulos de visão sobre um mesmo evento (por exemplo, o deslocamento de um animal ou a variação de temperatura em uma floresta), permitindo observar ambientes físicos ou monitorar equipamentos com elevado grau de precisão.

Com o avanço das tecnologias aplicadas às RSSFs, uma demanda é construir soluções de software que permitam integrar RSSF com outras redes sem fio e com a Internet de forma a possibilitar, por exemplo, o acesso remoto aos dados coletados pelos sensores por diferentes aplicações. Em cenários mais complexos, é possível projetar o desenvolvimento de mecanismos de colaboração para permitir que os dados coletados pelos sensores possam ser associados a outras informações de contexto, permitindo atuar sobre o ambiente em tempo real (por exemplo, dado o nível de umidade do solo e a previsão meteorológica atualizada, controlar dinamicamente a vazão do equipamento de irrigação).

O passo fundamental para alcançar essa integração das RSSFs é conseguir implementar a pilha de protocolos TCP/IP em dispositivos com severas restrições de recursos computacionais. As primeiras experiências de implementação da pilha TCP/IP em dispositivos de baixo consumo serviram para demonstrar a viabilidade desse esforço [Dunkels 2003, Hui and Culler 2008] e motivar a criação do grupo de trabalho 6LoWPAN (designado pela IETF para propor mecanismos e recomendações básicas para permitir a transmissão de pacotes IPv6 em redes de baixo consumo, entre as quais estão

as RSSFs). O protocolo IPv6 (sucessor do IPv4) tem sido adotado desde o início como protocolo de roteamento padrão para a integração das redes de baixo consumo (LoWPAN) com a Internet, pois oferece um amplo espaço de endereçamento e características adicionais que facilitam a compactação do seu cabeçalho, bem como a adoção de outras técnicas para redução da complexidade da sua operação em dispositivos com capacidades limitadas.

Entre as implementações existentes das recomendações do grupo 6LoWPAN para RSSFs, uma das mais importantes é a **BLIP** (Berkeley IP Information) [TinyOS-Blip 2012]. Em [Nascimento et al. 2011], os autores exploraram as possibilidades oferecidas pelo BLIP e mostraram como ligar os nós sensores com a Internet, usando o protocolo IPv6. Entretando, como ainda hoje a maioria dos computadores e subredes conectadas à Internet operam com o protocolo IPv4, é preciso construir soluções alternativas para permitir a interação remota entre computadores externos (operando em redes IPv4) e os nós sensores em uma RSSF.

Neste trabalho, apresentamos uma solução para essa questão permitindo interconectar os sensores de uma RSSF privada a computadores externos de forma transparente. Propomos uma solução baseada em VPN (*Virtual Private Network*), usando os protocolos IPSec [Kent and Seo 2005] e GRE [Farinacci et al. 2000] de forma integrada. Nosso objetivo principal é permitir que aplicações remotas possam interagir diretamente com os nós sensores da RSSF, requisitando dados sensoreados de forma segura.

O restante deste artigo está organizado da seguinte forma. Na seção 2 discutimos as tecnologias usadas para a criação de VPNs. Na seção 3 apresentamos a solução proposta de interação remota entre computadores externos e os nós sensores em uma RSSF. Na seção 4 mostramos os experimentos realizados. Finalmente, na seção 5 apresentamos as considerações finais do trabalho.

2. Tecnologias para criação de VPNs

VPN (*Virtual Private Network*) é uma forma de interligar computadores criando uma rede privada através de uma rede pública. No nosso caso, estamos interessados em permitir que um computador externo utilize a Internet (rede pública) para se comunicar com sensores remotos, como se esse computador estivesse dentro da RSSF particular (rede privada).

A solução proposta consiste em criar uma VPN interligando um computador externo — conectado a uma rede IPv4 — com a RSSF, de forma que seja possível a comunicação direta entre aplicações que executam nesse computador com a aplicação que executa nos nós sensores, como se todos estivessem na mesma rede. Para isso é preciso criar um túnel entre o computador externo e um computador que funciona como porta de entrada para a RSSF. O túnel fica responsável por despachar as informações de um lado a outro de forma transparente. A Figura 1 ilustra essa proposta.

O túnel pode ser realizado em várias camadas da pilha de rede, e geralmente o protocolo do túnel está na mesma camada ou abaixo do protocolo que está sendo encapsulado. O que ocorre é que as informações do procotolo encapsulado passam a ser o *payload* do protoloco de transferência do túnel. Podemos citar como exemplo pacotes IPv6 sendo "tunelados" por uma rede IPv4 (mesma camada), ou pacotes IP sobre IPX (camada abaixo). Como a transferência de dados se dará através da Internet, temos que garantir os requisitos mínimos de segurança na comunicação.

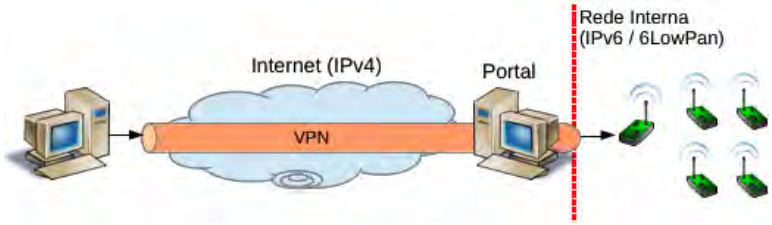

Figura 1. VPN com túnel IPv6/IPv4.

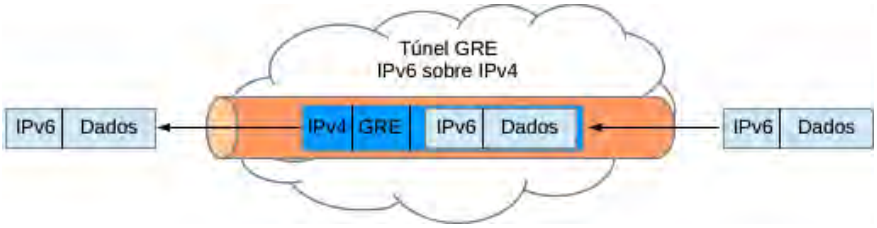

Figura 2. Túnel IPv6/IPv4 utilizando GRE.

2.1. GRE (*Generic Routing Encapsulation*)

GRE [Farinacci et al. 2000] permite o encapsulamento de um protocolo em outro. De forma geral, os dados do protocolo a ser encapsulado torna-se o *payload* do novo pacote. O GRE adiciona um cabeçalho extra contendo informações de controle e finalmente é adicionado ao pacote o cabeçalho do protocolo de transporte dos dados. A Figura 2 mostra o exemplo de um túnel IPv4 transportando pacotes IPv6.

O GRE é utilizado principalmente por sua flexibilidade. O cabeçalho necessita apenas de algumas informações de controle e o identificador do protocolo que está sendo transportado, o que gera um baixo *overhead*. Além disso, o GRE é *stateless*, ou seja, nenhum dos envolvidos necessita manter informações de controle do túnel. No entanto, uma desvantagem é que não há garantia nenhuma de autenticação, integridade ou confiabilidade da informação trafegada, pois o pacote original simplesmente é introduzido como *payload*.

2.2. IPSec

IPsec [Kent and Seo 2005] é uma extensão do protocolo IP que garante segurança nas comunicações em camada de rede. Foi inicialmente desenvolvido para o padrão IPv6 e depois portado para o IPv4.

Ele pode ser utilizado em dois modos: transporte ou túnel — Figura 3 (a) e (b), respectivamente. No modo transporte, apenas os dados do pacote IP original é encriptado ou autenticado. É útil quando apenas se deseja a segurança/autenticação da informação trafegada e é muito usado para comunicação entre duas máquinas. No modo túnel, todo o pacote IP original é encapsulado em um novo pacote IP. Ele permite portanto a criação de uma comunicação segura rede-a-rede, garantindo segurança inclusive dos endereços IPs originais de remetente e destinatário.

O IPsec disponibiliza os protocolos *Authentication Header* (AH) e *Encapsulating Security Payload* (ESP) para garantir autenticação, integridade e confidencialidade dos dados trafegados. O AH é usado para garantir a integridade da comunicação. Ele utiliza

Figura 3. Modo transporte (a) e modo túnel (b) do IPSec.

uma chave secreta para o cálculo de um código de autenticação baseado em *hash*, utilizando diversos campos do pacote IP original. Assim, se o pacote for modificado durante o trânsito, intencionalmente ou não, o receptor ao tentar calcular novamente o código *hash*, muito provavelmente encontrará um código diferente, e então poderá descartar o pacote. O ESP garante confidencialidade dos dados por meio de algoritmos de encriptação, e pode também garantir autenticação por meio de técnicas de *hash*, como no protocolo AH. Diversos algoritmos podem ser utilizados para a criptografia, porém os mais comuns são DES, tripe-DES, AES e Blowfish. Há ainda o algoritmo NULL que não encripta os dados, útil para depuração.

Para o estabelecimento do canal seguro, o IPSec precisa que os envolvidos troquem um conjunto de parâmetros. Assim, o estabelecimento da conexão (chamada de *Security Assossiation* (SA) foi divida em duas fases. Na primeira fase, é realizada a identificação dos dois pontos envolvidos e é criado um canal seguro utilizando, por exemplo, uma chave compartilhada entre eles. Esse canal seguro servirá para a fase dois, quando os envolvidos poderão então trocar os parâmetros para configurar o canal definitivo (geralmente os dados necessários para o funcionamento do AH e o ESP).

3. Solução com GRE e IPSec

Nossa solução tem como principais requisitos:

- Segurança: como a comunicação se dará pela Internet, é importante que os dados trafegados sejam confidenciais. Além disso, temos que garantir que somente o dono do experimento possa ter acesso, necessitando assim de um mecanismo de autenticação.
- Transporte por IPv4: apesar do protocolo IPv6 ser mais apropriado para RSSF, e já contando com implementações no Contiki e TinyOS, muitas instituições não dispõem desse tipo de conexão, recaindo no IPv4. Assim, decidimos focar no transporte de pacotes IPv6 sobre IPv4.

Nenhum dos protocolos descritos acima, atuando isoladamente, consegue preencher esses requisitos. O IPSec possui mecanismos para garantir o item de segurança, entretanto, segundo [Kent and Seo 2005], só é possível realizar associação entre endereços IPv4-IPv4 ou IPv6-IPv6, mas não IPv4-IPv6 — insuficiente para a segundo item.

Por outro lado, com GRE é possível transportar pacotes IPv6 na rede IPv4, mas não há nenhum suporte para segurança. A solução adotada neste caso foi a utilização dos dois protocolos em conjunto. Empregamos o IPSec na base para enviar de maneira segura pacotes IPv4 para o destino e garantir a autenticação na criação da VPN. Com o túnel IPv4 montado, utilizamos então o GRE para encapsular os pacotes IPv6 em IPv4, que serão então enviados via IPSec.

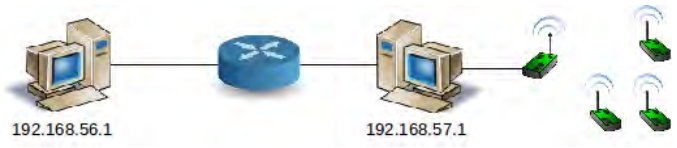

Figura 4. Arquitetura básica IPv4 para montagem de VPN via IPSec.

A seguir apresentaremos como foi realizada a configuração para acesso à rede. Usaremos como teste duas máquinas com Ubuntu 10.04LTS.

3.1. Configuração do IPSec

Na realização de nosso experimento de configuração, criamos um cenário com dois computadores em redes diferentes interligados por um roteador. Como mostrado na Figura 4, temos uma máquina com o endereço `192.168.56.1/24` que deseja acessar a RSSF, e a máquina que tem acesso à RSSF com o endereço `192.168.57.1/24`.

Primeiro, precisamos das ferramentas de controle do IPSec. Para isso temos que instalar os pacotes *ipsec-tools* e *racoon* nos dois computadores. Para a configuração da VPN é necessário modificar três arquivos. Apresentaremos a configuração para a máquina externa (`192.168.56.1`), e a configuração da outra é análoga, trocando apenas os endereços IPs.

O primeiro arquivo a ser configurado é o `/etc/ipsec-tools.conf`, que cria a política de comunicação de entrada e saída de pacotes entre as duas máquinas, dizendo que essa comunicação utilizará o protocolo ESP (para criptografia dos pacotes) em modo transporte:

```
1 spdadd 192.168.56.1 192.168.57.1 any -P out ipsec
2    esp/transport//require;
3 spdadd 192.168.57.1 192.168.56.1 any -P in ipsec
4    esp/transport//require;
```

Além de informar que a política de comunicação ponto-a-ponto entre as máquinas é segura, temos que configurar o sistema com os parâmetros para efetivamente criar esse canal seguro. Adicionamos então no arquivo `/etc/racoon/raconn.conf` a seguinte configuração:

```
1  path pre_shared_key "/etc/racoon/psk.txt";
2  remote 192.168.57.1 {
3          exchange_mode aggressive;
4          proposal {
5                  encryption_algorithm aes;
6                  hash_algorithm sha1;
7                  dh_group modp1024;
8                  authentication_method pre_shared_key;
9          }
10 }
11 sainfo address 192.168.56.1 any address 192.168.57.1 any {
12         pfs_group modp768;
13         encryption_algorithm aes;
14         authentication_algorithm hmac_md5;
15         compression_algorithm deflate;
16 }
```

Essa configuração define como serão realizadas as duas fases de estabelecimento de conexão do IPSec. A diretiva *remote* controla a primeira fase, dizendo que a conexão com a máquina conectada à RSSF utiliza criptografia AES, com hash SHA1 e o protocolo Diffie-Hellman (DH) — linhas 5, 6 e 7. O protocolo DH utiliza chave simétrica, e a linha 8 informa que essa chave deve ser buscada no arquivo de chaves compartilhadas, definido na linha 1. Como a mesma configuração será feita nas duas máquinas, ambas terão a mesma chave e o protocolo DH funcionará corretamente. A diretiva *sainfo* possui os parâmetros para a segunda fase do IPSec entre as duas máquinas. A linha 12 configura o PFS[1], que é uma propriedade de troca de chaves simétricas usada na conexão. Temos então a indicação do uso dos algoritmos AES e HMAC-MD5 nas linhas 13 e 14. Finalmente, a indicação de que não há compressão de dados, descrito na linha 15.

Por fim, temos que informar a chave no arquivo `/etc/racoon/psk.txt`. Essa chave pode ser indicada como uma sequencia de caracteres ASCII. No nosso caso, geramos um identificador aleatório e o utilizamos nas duas máquinas. Perceba que a chave está associada com o IP da máquina remota:

```
192.168.57.1   512fecfa-064c-43e8-8fda-94d0aba1d4ab
```

Temos que carregar nossa configuração executando os seguintes comandos em um *shell* como *root* — eles irão reiniciar os *daemons* responsáveis pela VPN.

```
# service setkey restart
# service racoon restart
```

3.2. Configuração do GRE

Com a conexão IPSec configurada, iremos criar um túnel IPv6-IPv4 entre as duas máquinas. Novamente, mostraremos a configuração de uma das máquinas, a configuração da outra é análoga. Assumindo o prefixo IPv6 `FEC0::/64` para a RSSF.

Primeiramente, temos que carregar o módulo do *kernel* chamado *ip_gre*, que é responsável pela gerência do túnel. Em seguida, vamos configurar o túnel, o que dá origem a uma nova interface virtual de rede no sistema (*gretun*) e a colocamos ativa. Finalmente, configuramos um endereço IPv6 que será utilizada na comunicação com a RSSF. Os comandos devem ser digitados no *shell* como *root*.

```
# modprobe ip_gre
# ip tunnel add gretun mode gre remote 192.168.57.1
  local 192.168.56.1 ttl 64 dev eth0
# ip link set gretun up
# ip -6 addr add 2000::1/64 dev gretun
```

A Figura 5 mostra a nossa rede virtual IPv6 sobreposta à rede IPv4 (não mostrada). A máquina ligada à RSSF servirá como um roteador, repassando os pacotes da RSSF para a máquina externa. Nesse caso, é interessante configurar na máquina externa (somente nela) a rota para a RSSF via `2000::2`:

```
# ip -6 route add FEC0::/64 via 2000::2
```

[1] *Perfect Forward Secrecy*: novas chaves simétricas não serão derivadas da chave da fase 1.

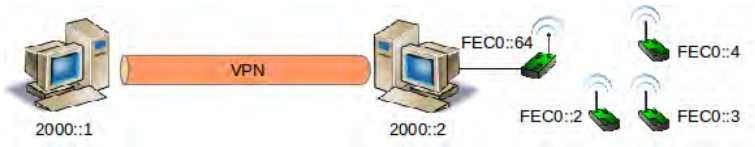

Figura 5. Túnel IPv6/IPv4 com GRE e IPSec para acesso à RSSF.

4. Avaliação

Para avaliar a nossa solução, criamos um cenário com três computadores (como ilustrado na Figura 4): um simulando a rede externa, outro com acesso à RSSF e o terceiro sendo usado como roteador. Realizamos toda a configuração já descrita, obtendo a configuração da Figura 5. Todos os computadores executavam Ubuntu 10.04 TLS.

Utilizamos a versão 1.0 do BLIP (pilha IPv6 para RSSF) que está disponível no TinyOS 2.1.1. O nó sensor com endereço FEC0::64 funciona como nó *sink*, e instalamos nos demais a aplicação de demonstração UDPEcho, disponível no TinyOS. Essa aplicação implementa o serviço Echo, que basicamente envia de volta o pacote UDP recebido na porta 7. Da máquina externa foi possível interagir com os nós sensores, enviando pacotes (contendo strings) com o programa *nc* (*netcat*), e recebendo-os de volta.

Em um exemplo mais complexo, utilizamos a aplicação apresentada em [Nascimento et al. 2011], que permite interagir com os sensores por meio de um navegador web. A aplicação foi desenvolvida para funcionar como um servidor web limitado, sendo executado no nó sensor, e permite efetuar a leitura dos sensores ou mesmo acender ou apagar alguns *leds* através de requisições HTTP. Para essa interação com os nós sensores foi utilizando o navegador Firefox, o qual estava sendo executado na máquina externa.

5. Considerações Finais

A interligação das RSSFs com a Internet começa a virar realidade com a utilização do protocolo IPv6 por parte dos nós sensores. Por outro lado, não temos ainda a adoção desse protocolo em massa na Internet, impossibilitando o acesso direto às RSSFs executando em IPv6. Devemos então utilizar soluções transitórias, por exemplo, o protocolo IPv4 para transportar IPv6. Além disso, nem toda RSSF deve ser acessível publicamente. Assim como algumas redes coorporativas, que só liberam o acesso a seus serviços via VPN, podemos ter cenários de RSSF privadas (por exemplo, “casas inteligentes”) onde o acesso através da Internet seja possível, mas controlado.

Apresentamos neste trabalho uma solução baseada em IPSec e GRE para acesso à RSSF considerada privada. A utilização dos dois protocolos foi necessária pois ambos, isoladamente, não possuíam os requisitos para disponibilizar o acesso remoto. Utilizando o IPSec foi possível criar um canal seguro entre a máquina remota e um computador com acesso à RSSF. Esse canal, além de manter a comunicação criptografada, utiliza uma chave para estabelimento da conexão, o que permite controlar quem está requisitando acesso. Já o GRE foi empregado para efetuar o transporte de pacotes IPv6 sobre IPv4.

A configuração do GRE não apresenta grande dificuldade, no entanto, o protocolo IPSec é bem mais complexo. O IPSec é divido nos protocolos AH e ESP, que oferecem

várias funcionalidades, sendo necessário diversos parâmetros de configuração (por exemplo, modo de funcionamento, algoritmos de criptografia, compressão e chaves). A curva de aprendizado se mostrou muito alta, ainda que, em nossa opinião, exploramos pouco os protocolos. Além disso, a configuração sempre se baseia em modificações de arquivos de configuração e recarga de *daemons*. A construção de sistemas, seja via web ou desktop, seria mais simples se houvesse uma forma mais simples de interagir com o SO, ou seja, uma API de mais alto nível para a programação.

O ponto positivo dessa solução é a utilização de protocolos padronizados, por outro lado, a configuração do IPSec pode se tornar cada vez mais complexa dependendo do cenário que ele for empregado. Gostaríamos de avaliar, em trabalhos futuros, outras alternativas como o OpenVPN [OpenVPN Inc. 2012] ou o PPTP [Hamzeh et al. 1999].

6. Agradecimentos

Este trabalho foi desenvolvido como parte do projeto CIA2[2], financiado pela Rede Nacional de Ensino e Pesquisa (RNP).

Referências

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.

Estrin, D., Culler, D., Pister, K., and Sukhatme, G. (2002). Connecting the physical world with pervasive networks. *Pervasive Computing, IEEE*, 1(1):59–69. http://doi.ieeecomputersociety.org/10.1109/MPRV.2002.993145.

Farinacci, D., Li, T., Hanks, S., Meyer, D., and Traina, P. (2000). RFC2784: Generic routing encapsulation (GRE). `http://tools.ietf.org/html/rfc2784`.

Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W., and Zorn, G. (1999). RFC-2637: Point-to-point tunneling protocol (PPTP). `http://www.ietf.org/rfc/rfc2637.txt`.

Hui, J. and Culler, D. (2008). IP is dead, long live IP for wireless sensor networks. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pages 15–28. ACM Press.

Kent, S. and Seo, K. (2005). RFC4301: Security architecture for the internet protocol. `http://tools.ietf.org/html/rfc4301`.

Nascimento, M., Silvestre, B., and Zenha, L. (2011). Interligando RSSFs com a Internet utilizando IPv6. In *Encontro Anual de Computação*.

OpenVPN Inc. (2012). OpenVPN. `http://www.openvpn.net`.

TinyOS-Blip (2012). Blip — Berkeley IP Information. `http://smote.cs.berkeley.edu:8000/tracenv/wiki/blip`.

Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52:2292–2330.

[2]Construindo Cidades Inteligentes: da instrumentação dos ambientes ao desenvolvimento de aplicações — http://www.nr2.ufpr.br/∼cia2