Análise de Desempenho do Protocolo DTLS para Internet das Coisas

Daniele F. de Jesus e João H. Kleinschmidt

Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas (CECS)
Universidade Federal do ABC (UFABC) - Santo André – SP – Brasil

{daniele.freitas,joao.kleinschmidt}@ufabc.edu.br

Resumo. A Internet das Coisas (IoT) se refere à interconexão da rede formada por objetos inteligentes, além do conjunto das tecnologias que compõem esses objetos e suas aplicações e serviços. Possui grandes limitações em recursos computacionais, o que torna desafiadora a questão de segurança dos dados. Este trabalho analisa o desempenho do protocolo de segurança DTLS em ambientes IoT. Foi utilizado o ambiente de simulação *Contiki* para a análise de consumo de energia e tempo de resposta em uma rede cliente/servidor com DTLS. Os resultados obtidos mostram que o DTLS é uma solução viável para ser utilizado na IoT e que pequenas alterações no protocolo padrão podem melhorar o desempenho.

1. Introdução

A Internet das Coisas (ou IoT, do inglês *Internet of Things*) possibilita a interconexão de objetos inteligentes com a rede tradicional [Palatella et al 2013]. Os protocolos existentes para a Internet tradicional, como HTTP (*Hyper Text Transfer Protocol*), TCP (*Transport Control Protocol*) e IP (*Internet Protocol*) não consideram as características de dispositivos com poucos recursos de memória, processamento e energia, como é caso da IoT. Entretanto, vários novos protocolos, como CoAP (*Constrained Application Protocol*) na camada de aplicação e 6LoWPAN (*IPv6 Over Low Power Wireless Personal Area Network)* na camada de rede já foram padronizados para substituir protocolos como HTTP e IP na IoT [Palatella et al 2013]. Um problema que permanece em aberto é o protocolo de segurança a ser utilizado. Vários autores têm proposto o uso do protocolo *Datagram Transport Layer Security* (DTLS) como alternativa [Keoh et al 2014]. O DTLS foi desenvolvido para ser usado com o protocolo de transporte UDP (*User Datagram Protocol*). Este artigo analisa o desempenho do protocolo DTLS em uma rede IoT. Alguns trabalhos recentes analisam o uso de DTLS e propõe algumas modificações no protocolo [Keoh et al 2014]. Neste trabalho será analisado o desempenho do DTLS e algumas variações [Raza et al 2013] [Hartk e Bergmann 2012], que não foram analisadas em conjunto na literatura.

2. Protocolo DTLS

O DTLS é um protocolo de segurança que automatiza o gerenciamento e a distribuição de chaves, além de automatizar a autenticação e a encriptação dos dados, utilizado para proteção das mensagens trafegadas via UDP para aplicações cliente/servidor [Rescorla e Modagugu 2006]. O *handshake* do DTLS é organizado em *flights*, usado para negociar chaves de segurança, conjuntos de codificação e métodos de compressão. Só após o handshake é que cliente e servidor podem enviar dados. Na sua

forma completa, o *handshake* possui seis *flights*, como mostra a Figura 1. Opcionalmente, o DTLS pode usar certificados digitais no cliente e servidor. Neste trabalho não serão usados certificados.

Uma das propostas na literatura para melhorar o desempenho é o *handshake* reduzido [Hartk e Bergmann 2012]. O *handshake* padrão tem dois *flights* adicionais quando comparado com o comumente utilizado TLS, resultante da adição de uma troca de *cookies* sem estado. Esta troca é projetada para evitar certos ataques de negação de serviço. No entanto, isto pode ser reduzido em um datagrama, permitindo que um *handshake* completo possa ser encurtado para quatro *fligths* (ou seja, retirar os *fligths* 2 e 3, incluindo os serviços destes no *flight* 1). Outra proposta para adaptar o protocolo DTLS ao contexto da IoT foi apresentada por [Raza et al 2013], que basicamente é uma compressão do cabeçalho DTLS para reduzir a quantidade das informações trafegadas. Nesse trabalho recebeu a notação de DTLS comprimido.

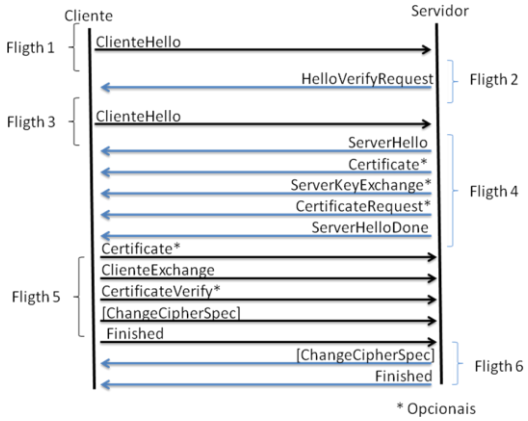

Figura 1 - *Flights* do *Handshake* no DTLS

3. Análise de Desempenho do Protocolo DTLS

Para análise de desempenho do protocolo DTLS foi utilizado o Sistema Operacional Contiki e o simulador Cooja [Contiki 2015], que vem sendo amplamente utilizado em sistemas IoT. As simulações utilizam dois nós em uma arquitetura cliente/servidor (Figura 1). Foi utilizada a seguinte pilha de protocolos: CoAP, DTLS, UDP, 6LoWPAN e IEEE 802.15.4. O modelo do canal de rádio é o UDGM (*Unit Disk Graph Model*). O dispositivo simulado é o *Wismote* (16 MHz, MSP430, microcontrolador 16-bit RISC, 128/16 kB de ROM/RAM e transceptor IEEE 802.15.4), por ser mais robusto em memória e processamento, necessários para a utilização do DTLS. O algoritmo de criptografia utilizado é o AES e o tamanho de chave é de 128 bits.

A Figura 2 apresenta a média de tempo do *handshake* para cliente e servidor, separadas por *fligths*. É possível verificar que tanto no cliente como no servidor, os *fligth* 4, 5 e 6 são os que duram mais tempo, por causa das definições e troca de chaves, ou seja, no estabelecimento da comunicação segura. O uso do DTLS comprimido apresenta um desempenho muito próximo ao DTLS padrão.

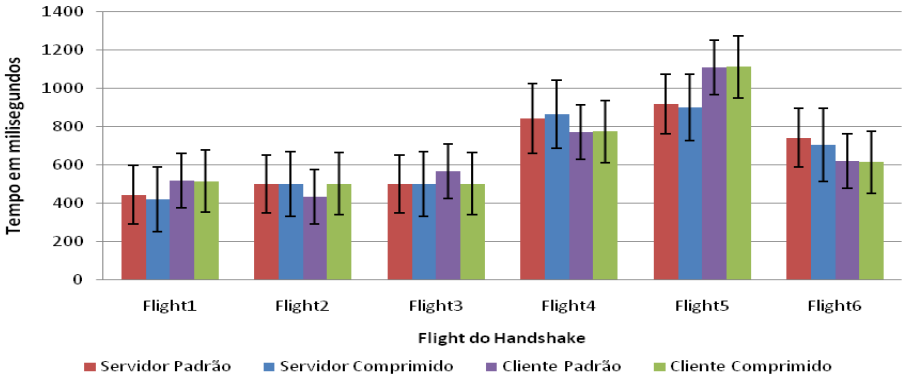

Figura 2 - Tempo total do *handshake* por *flights*

Na Figura 3 são comparados o consumo de energia do cliente e servidor por *flight* do *handshake*. No cliente com DTLS padrão, o pico de consumo ocorre no *flight* 4, fase em que ele recebe os dados que serão utilizados na definição das chaves. Já no servidor padrão, esse pico ocorre no *flight* 5, fase em que ele define a chave a ser utilizada na comunicação segura e recebe a confirmação do cliente nesse processo, para solicitar o fechamento do *handshake*, que ocorre no *flight* 6. No servidor com DTLS padrão o consumo de energia se mantém maior que o DTLS comprimido em todos os *flights*. Portanto, no processo de *handshake*, o DTLS comprimido apresenta um melhor desempenho em relação ao consumo de energia (Figura 3), ainda que não tenha influência significativa no tempo (Figura 2).

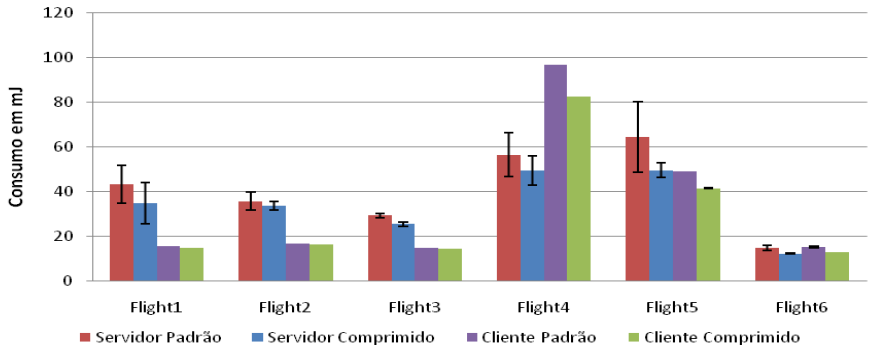

Figura 3 - Consumo de Energia por *flight*

A Figura 4a mostra o tempo total do handshake para cliente e servidor, usando o DTLS padrão, DTLS comprimido e o *handshake* reduzido. Ao se utilizar o *handshake* reduzido o tempo fica aproximadamente 25% menor que no *handshake* completo. Isso por que foram retirados dois *fligths*. Em relação ao consumo total de energia (Figura 4b), com o *handshake* completo no servidor, o formato comprimido é em média 16% menor em relação ao formato padrão, e no cliente o formato comprimido é 12% menor que o formato padrão. Já em comparação ao *handshake* reduzido, no servidor a redução é de 26% em relação ao DTLS padrão e de 28% no comprimido; já no cliente a redução é de 15% em relação ao DTLS padrão e de 16% no comprimido. Como esperado, ao se reduzir os *flights* no *handshake*, o tempo e consumo de energia total diminuem, sugerindo que o uso de handshake reduzido e compressão do DTLS podem trazer ganhos significativos de desempenho.

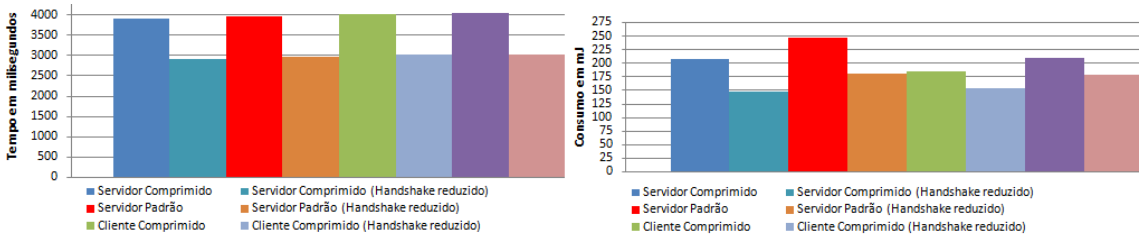

Figura 4. (a) Tempo total do *handshake* **(b) Consumo total de energia do *handshake***

Outra análise feita foi o envio de um pacote de dados entre cliente e servidor, após o estabelecimento da conexão segura. O cálculo de consumo de energia é iniciado a partir de estabelecido o *handshake.* A Figura 5(a) mostra o consumo de energia com DTLS padrão, DTLS comprimido e sem DTLS, para uma requisição do cliente ao servidor e envio de um pacote de dados pelo servidor. Não utilizar segurança (sem DTLS) consome menos da metade da energia comparado ao uso do protocolo DTLS. O tempo para envio e recebimento de um pacote de dados (requisição do cliente para o servidor) é o RTT (*Round Trip Time*). O tempo RTT para um cliente sem DTLS teve uma grande diferença em relação a um cliente com DTLS, como mostra a Figura 5b. O RTT aumenta com o uso de DTLS padrão e DTLS comprimido, por causa da criptografia das mensagens de requisição e resposta entre cliente e servidor.

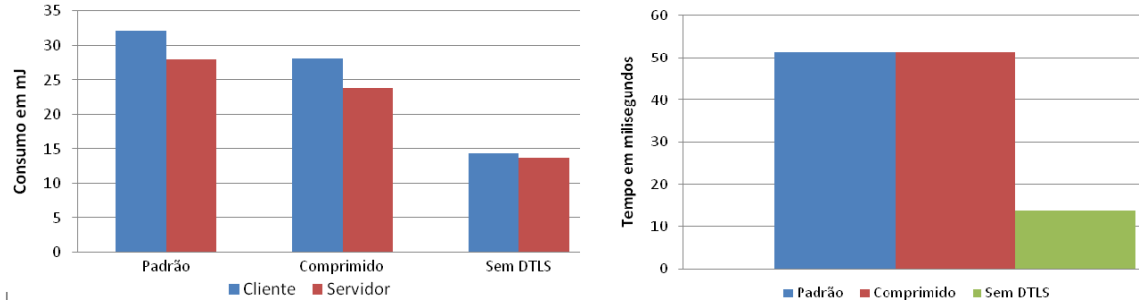

Figura 5. (a) Consumo de energia para um pacote de dados **(b) Tempo total (RTT) no cliente**

4. Considerações Finais

Os resultados obtidos mostram que a compressão do cabeçalho no DTLS reduz o consumo de energia na rede, o que torna esse formato recomendado para uso nas redes IoT. O *handshake* reduzido apresenta bons resultados em relação ao tempo para estabelecer uma conexão segura, reduzindo também o consumo de energia. Estas duas propostas são bastante promissoras para serem utilizadas na IoT. Os trabalhos futuros incluem verificar o impacto do uso de certificados no DTLS e analisar outras arquiteturas e cenários de rede, como uma arquitetura orientada a serviços.

Referências

Contiki Operating System for the Internet of Things (2015). Disponível em: www.contiki-os.org

Hartke, K; Bergmann O. (2012) DTLS in Constrained Environments. Draft-hartke-core-codtls-01. Internet Engineering Task Force (IETF).

Keoh, Sye Loong; Kumar, Sandeep S.; Tschofenig, Hannes (2014). Securing the Internet of Things: A Standardization Perspective. IEEE Internet of Things Journal, Vol. 1, NO. 3, June 2014.

Palattella, Maria Rita, Et Al. (2013) Standardized Protocol Stack for the IoT. IEEE Communications Surveys & Tutorials, Vol. 15, No. 3, Third Quarter.

Raza, Shahid; Et Al. (2013). Lithe: Lightweight Secure CoAP for the Internet of Things. IEEE Sensors Journal, vol. 13, no. 10, October 2013.

Rescorla, E.; Modadugu, N. Datagram Transport Layer Security. (2006). RFC 4347: Datagram Transport Layer Security. Internet Engineering Task Force (IETF). 2006.

