Análise Crítica à Plataforma Contiki

José António Portela Areia

Outubro 2022

Introdução

Contiki é um sistema operativo focado em redes de pouco consumo com objetivo principal de servir dispositivos IoT que, juntamente com a ferramenta Cooja, permite simular um cenário virtual com vários protocolos de comunicação a eles inerentes.
Em teoria, esta plataforma seria ideal para a representação de um cenário simulado para o projeto em questão, no entanto, durante um desenvolvimento com duração de 4 meses foram encontradas várias falhas e/ou problemas que culminaram na desistência da utilização da mesma.
Neste relatório irão ser analisados e explicados em detalhe todos os problemas que surgiram nos diferentes tipos de implementação abordados durante o desenvolvimento do projeto, que refletem todo o trabalho desenvolvido ao longo do período temporal em epígrafe.

1. Plataforma Contiki-NG (Native Linux) & Cooja

Após se ter um cenário pensado e desenhado, a próxima fase consistiu-se na escolha da ferramenta a utilizar para representar o cenário e, após um vasta pesquisa à procura de uma solução ideal, optou-se pela plataforma Contiki-NG.

A plataforma escolhida é a versão mais recente do sistema operativo Contiki, que oferece diversas instalações de modo a suportar vários sistemas diferentes. Neste caso, para a plataforma Contiki-NG, testaram-se as instalações Docker e Native toolchain (Linux).

A ferramenta Cooja, disponibilizada através da instalação da plataforma Contiki-NG, oferece uma '*interface'* gráfica que permite a simulação de uma rede e a possibilidade de criação de vários '*motes',* que representam, de uma forma simulada, equipamento físico dedicado para redes IoT como, por exemplo, os dispositivos NRF52 ou Zolertia Zoul.

Numa primeira abordagem, recorreu-se à instalação nativa da plataforma Contiki-NG numa máquina Linux (Debian 11 Bullseye) e à utilização da ferramenta Cooja, onde foram encontradas as primeiras limitações. Essas limitações refletem-se no uso dos protocolos de comunicação propostos para o projeto (CoAP e MQTT).

1.1. Protocolo CoAP

A plataforma Contiki-NG não oferece um cliente que comunique de uma forma estável com o servidor e, usando o protocolo CoAP, torna-se impossível associar um '*mote'* a um cliente CoAP na ferramenta Cooja, o que se revela uma limitação flagrante. No entanto, é possível criar um cliente CoAP compilando-o de forma nativa, embora não fosse o ideal para a representação do cenário, visto que não permite ter vários clientes CoAP numa comunicação simultânea com o servidor.

Durante a procura de suporte na documentação disponibilizada pela plataforma Contiki-NG, encontrou-se um exemplo, "*We have no tutorial for a Contiki-NG CoAP client yet, but we do provide an example firmware under examples [...]*" e denotou-se que o exemplo disponibilizado escasseava de suporte e documentação, sendo difícil a adaptação do mesmo para este projeto.

1.2. Protocolo MQTT

Dentro da plataforma Contiki-NG também existe a possibilidade da implementação do protocolo MQTT que, no entanto, não oferece alguma configuração possível: "*The MQTT engine is implemented [...]. The implementation does not currently offer any configuration options*" e evidencia-se a ausência de um '*broker'* MQTT, sendo que era preciso um '*broker*' externo para alcançar a comunicação necessária. Sublinha-se ainda que a comunicação nem sempre era possível, tendo apenas sido alcançada uma vez nas várias tentativas feitas.

Terminando a análise do protocolo MQTT, a plataforma refere três limitações (encontradas previamente) em termos de falta de suporte, nomeadamente para MQTT QoS

2, para a receção e publicação de mensagens com QoS 1 e da implementação de MQTT sobre TLS.

2. Plataforma InstantContiki 3.0 & Cooja

Perante as limitações previamente expostas, optou-se por outra plataforma, também disponibilizada pela entidade Contiki, utilizando uma máquina virtual preparada para o efeito.

A máquina virtual continha tudo o que era necessário para trabalhar com o Contiki na sua versão 3.0 (versão anterior à utilizada na primeira abordagem) e o sistema operativo definido foi o Ubuntu Desktop 14.04 LTS (lançado em 2014), o que se traduzia em '*software'* desatualizado na sua maioria. No entanto, optou-se por esta solução, visto que, teoricamente, suportava a implementação de ambos os protocolos (CoAP e MQTT), mas, na prática, também foram encontradas limitações na sua implementação.

2.1. Protocolo CoAP

O protocolo CoAP revelou-se uma limitação, visto que não estava disponível uma implementação "pura" deste protocolo, apenas uma implementação genérica de uma aplicação REST cliente/servidor. Por isso, não foi possível obter uma visualização correta (através da extensão '*Copper'*) com o intuito de verificar os parâmetros enviados através do protocolo.

2.2. Protocolo MQTT

Não estando disponível uma implementação deste protocolo nesta versão do Contiki, foi possível implementar um protocolo similar ao MQTT (MQTT-SN) através de código-aberto. O protocolo MQTT-SN funcionou corretamente, com os clientes representados por '*motes'* na ferramenta Cooja e a existência do '*broker'* externo, com auxílio da ferramenta '*Mosquitto'*.

Após se ter colocado estes dois protocolos a comunicar corretamente, dentro das possibilidades e limitações apresentadas, passou-se para a fase seguinte, de recolha tráfego durante ataques sobre o cenário outrora simulado.

Durante a recolha de tráfego, notou-se que o tráfego do protocolo "CoAP" não foi recolhido, dúvida sobre a qual permanece até ao momento. No entanto, decidiu-se continuar e prosseguir para a fase de ataques sobre o cenário simulado. Para o efeito, utilizou-se três ferramentas distintas, nomeadamente a framework *RPL-Attacks,* a *'toolkit' THC-IPv6* e a *MQTT-Malaria,* ferramenta focada em ataques para o protocolo MQTT.

Duas das ferramentas referidas (*RPL-Attacks* e *MQTT-Malaria*) não foram instaladas e/ou configuradas com sucesso, devido ao facto exposto no início deste capítulo (o sistema operativo usado limitava a sua instalação e/ou configuração devido a dependências em falta). No entanto, foram executadas várias tentativas para a colocação

correta destas ferramentas através, por exemplo, da instalação das várias dependências em falta, resultando em erros esdrúxulos.

A última ferramenta disponível (*'toolkit' THC-IPv6*), que visava fazer vários ataques de DoS, tinha como alvo principal uma '*interface*' específica e não um endereço IPv6, o que consistia numa limitação para o cenário atual, visto que os '*motes*' eram identificados por um endereço IPv6 e não por uma '*interface*'. No entanto, decidiu-se realizar um '*flood-attack*' sobre a '*interface*' TUN0, desenvolvida para suportar a comunicação exterior com o '*broker*' MQTT-SN, e, após a recolha do tráfego gerado, consta-se que não surtiu o efeito esperado.

3. Plataforma Contiki-NG (Docker) & Cooja

Após as duas tentativas descritas, optou-se ainda pela a instalação e configuração da plataforma Contiki-NG através do Docker, sendo uma versão mais recente e estável da aplicação, e, em teoria e segundo uma parte descrita na documentação: "*We recommend using the Contiki-NG Docker image for easy setup of a consistent development environment*", permitiria fazer o pretendido.

Seguindo a documentação oficial, procedeu-se a uma instalação normal num ambiente Docker da plataforma Contiki-NG e tentou-se iniciar a ferramenta Cooja, a qual gerou o primeiro erro, não sendo possível concluir a sua inicialização, que disponibilizaria um cenário simulado para o projeto. Após alguma leitura e investigação chegou-se à conclusão que teriam mudado de tecnologia para o arranque da ferramenta, sem que constasse na documentação oficial.

Procedeu-se, mais uma vez, com o plano inicial de recriar o cenário neste novo ambiente, mas ao tentar compilar o *RPL Border Router* – router que faria a ligação do cenário simulado a um ambiente exterior – foi apresentado um erro de compilação, que explicava que se deve compilar de uma forma nativa e não associado a um '*mote*' do tipo Z1 e/ou Sky, constante na documentação oficial. Porém, seguindo as instruções do programa, não foi possível resolver o problema. Investigou-se, despendeu-se tempo, mas não se encontrou uma solução para este problema fulcral que impediu o avanço do projeto.

Conclusão

Acredita-se que a plataforma Contiki-NG, juntamente com a ferramenta Cooja, oferece grandes ambições e possibilidades para trabalhos excelentes na área das redes IoT, no entanto, é preciso que referir que tais possibilidades não se enquadram no projeto a desenvolver, não suportando na totalidade os protocolos pretendidos, bem como os ataques que posteriormente eram necessários serem realizados.

Junta-se ainda que, no percurso destes 4 meses, foi realizado um trabalho exaustivo para solucionar todos os entraves que foram aparecendo, para culminar na solução perfeita para o projeto em questão, mas, infelizmente, chegou-se a um ponto em que não foi possível avançar mais, tendo em conta que existem outras soluções viáveis que não requerem o '*know-how*' que esta plataforma requer, bem como o tempo despendido na procura de soluções para todos os problemas encontrados.

Finalizando, a plataforma oferece bons recursos para uma simulação e aplicação em equipamento físico IoT, mas não oferece o necessário para o projeto a realizar.