IPD438 - Seminario de Redes de Computadores
Internet de las Cosas y Seguridad

Preparado por
Javier Romero Schmidt, ROL: 201121004-2
e-mail: javier.romeros@alumnos.usm.cl

17 de noviembre de 2016

Resumen

La Internet de las Cosas o IoT, por su nombre en Inglés Internet of Things, es un campo de estudio emergente. Esta se trata de la interconexión de toda clase de dispositivos a la red de internet, desde sensores y actuadores a maquinaria, computadoras y automóviles, toda clase de objetos que puedan aprovechar las ventajas y beneficios que la conexión a una red pueden traer. Pero esto también conlleva a ciertas debilidades que hay que enfrentar y nuevas dificultades que en otra clase de equipos no se ven afectados. En este proyecto de curso se estudiará un protocolo ligero estandar llamado CoAP (Constrained Application Protocol), un protocolo de comunicación a nivel de aplicación diseñado para su uso en dispositivos con capacidades restringidas. Además se estudiarán las posibilidades de implementar mecanismos de seguridad en este, como DTLS (Datagram Transport Layer Security), y mecanismos de encriptación ligeros.

1. Introducción

La Internet de las Cosas, o IoT, es un concepto propuesto por Kevin Ashton en una charla en el año 1997[1]. Este concepto se refiere a la interconexión, ya sean cotidianos o industriales, etcétera, a la red de Internet. El objetivo de esto es aprovechar las ventajas que tiene el uso de Internet en el uso de las cosas, ya sea para monitoreo, control, comodidad, y sin fin de otros motivos que solo están limitados por las capacidades de la tecnología y la creatividad del diseñador.

Pero esto no es solo beneficios ya que hay que tener en consideración algunos puntos importantes. Los dispositivos deben ser pequeños y de bajo consumo, de forma de aprovechar la posibilidad de utilizar una gran cantidad de estos, lo que lleva a una consecuencia lógica, tienen pocas capacidades energéticas. En IoT es imprescindible que el consumo energético sea muy bajo, lo que significa que se debe ahorrar potencia en todas sus partes, incluso en la comunicación. Así surgen protocolos como CoAP definido por el estándar [RFC 7252].

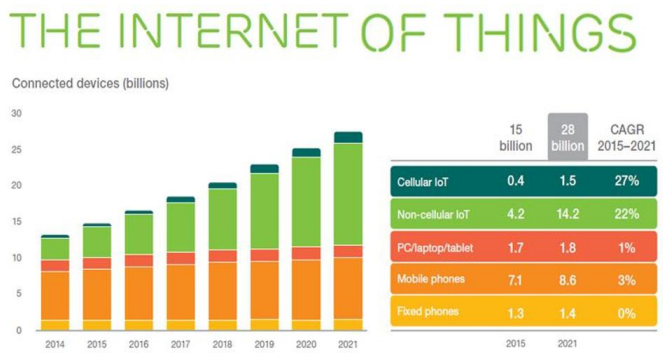

Figura 1: Estimaciones de la cantidad de dispositivos IoT en el mundo[3].

Lo que nos lleva a la siguiente pregunta lógica, es este estándar capaz de implementar mecanismos de seguridad. En un principio se decía que para el año 2020 la cantidad de dispositivos IoT iba a alcanzar la cifra de 50 millones, hoy se habla de 20 a 30 millones[2] lo que aún sigue siendo una cifra monstruosa. Esto conlleva a una variedad de distintos dispositivos, de distinta naturaleza interconectados y enlazados a Internet, lo que trae vulnerabilidad a la seguridad a una mayor escala. Para se propone a la implementación de protocolos de seguridad como DTLS, el cual se explicará en más detalle en este documento, pero también aparecen técnicas de cifrado ligeras, que buscan lograr tener confidencialidad de la comunicación a menor costo energético.

Es fundamental el tema de la seguridad ya que las vulnerabilidades ya no recaen sobre temas conocidos, como la denegación de servicios o el robo de información, sino que aparecen más y nuevos problemas como los ataques por denegación de sueño (denial of sleep attacks), que buscan drenar las baterías de los dispositivos.

Es por esto que en un campo emergente como lo es IoT, la seguridad aparece como un tema de interés para la investigación, notando además que no es realmente un tema nuevo ya que los protocolos de comunicación sobre redes existen y se han investigado desde hace años, lo que nos

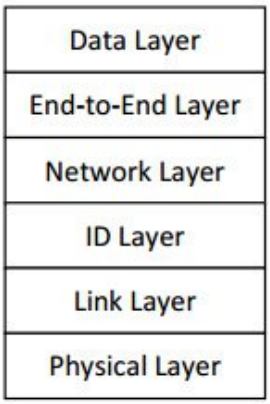

Figura 2: Pila de Comunicación de IoT.

permite aprovechar este conocimiento en este nuevo campo. Este documento esta estructurado de la forma que sigue: Sección 2 se hablará de protocolos de comunicación, centrado alrededor de CoAP, en la Sección 3 se estudiará la seguridad en este protocolo, en la Sección 4 se estudiará una herramienta para la implementación y simulación de redes IoT, y en la Sección 5 se dará una conclusión.

2. Protocolos de Comunicación

Una de las características de IoT es el uso de dispositivos de bajo consumo, ya que el objetivo es dar valor agregado y incluir funcionalidades a objetos, sin tener un costo mayor. Es por esto que un protocolo de comunicación "comprimido" se hace necesario. Así nacen algunos protocolos para la comunicación entre dispositivos en IoT.

Uno de los tantos modelos que describen las capas de comunicación en IoT se puede observar en Fig. 2[4]. Este busca emular el modelo OSI pero adaptado a las necesidades de IoT. En esta sección nos centraremos en la capa de data, con el protocolo CoAP, existen otros para esta misma capa como MQTT o SMCP.

CoAP, o llamado Constrained Application Protocol, definido en el estándar [RFC 7252][5], busca definir un protocolo para comunicación dispositivo a dispositivo a nivel de capa de aplicación. CoAP en sí está diseñado para reemplazar a HTTP en dispositivos de capacidades restringidas.

Sus capacidades principales consisten en:

- Fácilmente traducible a HTTP para integración con la Web.
- Capacidad de multidifusión de mensajes.
- Cabecera pequeña.
- Baja complejidad de análisis sintáctico.
- Soporte de URI y tipo de contenido.

Ver	T	TKL	Code	Message ID
Token (if any, TKL bytes) ...				
Options (if any) ...				
1 1	1 1	1 1 1 1	Payload (if any) ...	

Figura 3: Estructura de Mensaje de CoAP.

Esto permite tener un protocolo de bajo costo computacional y menor uso de memoria, por lo que se pueden utilizar dispositivos con menores recursos. Normalmente CoAP es transportado sobre UDP, pero también se puede realizar sobre otros protocolos. La estructura de un mensaje CoAP se puede observar en Fig. 3. Si se quiere estudiar más sobre este protocolo se debe dirigir a la página Web de la IETF donde se puede encontrar la documentación respectiva de este estándar, donde se podrá encontrar información desde las capacidades del protocolo hasta algunas de las consideraciones con respecto a la seguridad de este.

Existen diversas implementaciones de este protocolo en diferentes lenguajes, como Californium[6] en Java, CoAP.NET[7] en C#, Ruby coap[8] en Ruby, y muchos otros. Pero como implementar seguridad sobre el protocolo.

Diversos papers han estudiado como implementar seguridad sobre este protocolo de comunicación, Rahman et. al[9] analizan la seguridad en protocolos IoT enfocado en CoAP. Autores como Capossele et. al[10] proponen un método para implementar CoAP sobre DTLS, protocolo de capa de transporte, de una forma más óptima y con menos consumo de recursos, o Park et. al[11] proponen un método de seguridad ligera por medio de un handshake delegado de DTLS. En general se ha estudiado utilizar DTLS en capa de transporte para proveer de seguridad a la comunicación, pero su integración es lenta y consumidora de recursos, por lo que se tienen modificar de forma que pueda ser una solución real en el ámbito de IoT. En la siguiente sección estudiaré el uso de este protocolo para seguridad.

3. Seguridad en IoT

Como he explicado previamente el crecimiento de la cantidad de dispositivos IoT que ha habido en los últimos años y el que va avenir, trae consigo a su vez un enorme aumento en la cantidad de información que circula a través de Internet. Información que se debe proteger de ataques externos, al igual que proteger el acceso o interrupción indebidos de los mismos dispositivos. Así es como surge el estudio de métodos para proveer de seguridad a dispositivos con condiciones de funcionamiento restringidas, que son la mayor parte de IoT, como clusters de pequeños sensores o dispositivos de bajo consumo con memorias de reducido tamaño.

En la sección anterior se dijo que muchos autores han propuesto el uso de DTLS en capa de transporte para entregar seguridad a la IoT, el cual tiene las características

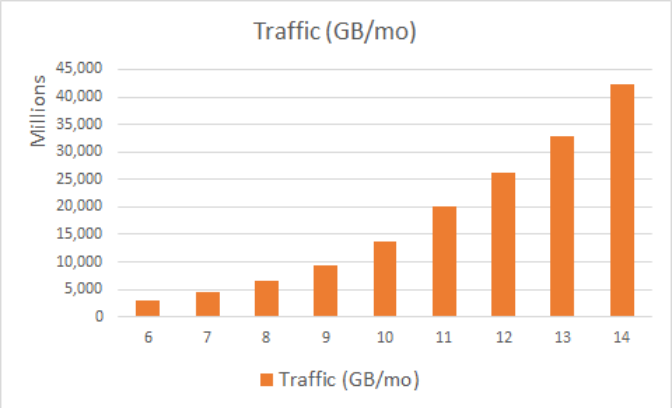

Figura 4: Crecimiento del tráfico de datos de Internet[12].

de proveer confidencialidad, integridad y autenticidad en la comunicación, los tres pilares fundamentales de la seguridad de un mensaje en una red. Pero este protocolo tiene algunas desventajas para ser utilizado en este campo, esta poco optimizado para su uso en dispositivos de capacidades restringidas, a pesar de su gran difusión.

Capossele et. al[10] propone utilizar este protocolo, que es una versión de TLS basada en UDP, ya que está diseñado para proveer una asociación segura entre punto y punto. Además permite negociar servicios de seguridad y mecanismos de encriptación, lo que a su vez significa flexibilidad. Los autores se aprovechan de la arquitectura de CoAP que proveen de una conexión orientada a la comunicación y de la fragmentación que se puede realizar en base a la transmisión basada en bloques definida por el protocolo. En Fig. 5 se puede ver como se establece la conexión, con esto se tiene una comunicación segura en base al método de encriptación usado.

A pesar de que el handshake de DTLS es un método que puede ser en ciertos dispositivos muy complejo y necesitar muchos recursos para almacenar la información de seguridad, es un buen acercamiento para un campo aún emergente, ya que es un estándar conocido y con múltiples implementaciones.

Algunos de los resultados obtenidos son bastante satisfactorios. Para evaluar, implementar y optimizar su propuesta los autores proponen realizar sus pruebas en un MagoNode, plataforma de IoT de bajo poder y con recursos limitados, el cual posee capacidades de comunicación y memoria suficiente para implementar TinyOS con protocolos 6LoWPAN (estándar para transmisión de paquetes IPv6 por en redes 802.15.4), RPL (protocolo de ruteo para IPv6 en redes de bajo consumo), UDP, CoAP y la implementación de DTLS. Encontrando que es posible disminuir el overhead generado por DTLS sobre CoAP en 236B comparado con una implementación estándar del protocolo, además de reducir el uso de ROM en cerca de 23 %.

Finalmente en la siguiente sección se estudiará una herramienta muy útil para la simulación e implementación de redes de dispositivos restringidos, el sistema operativo de código abierto Contiki.

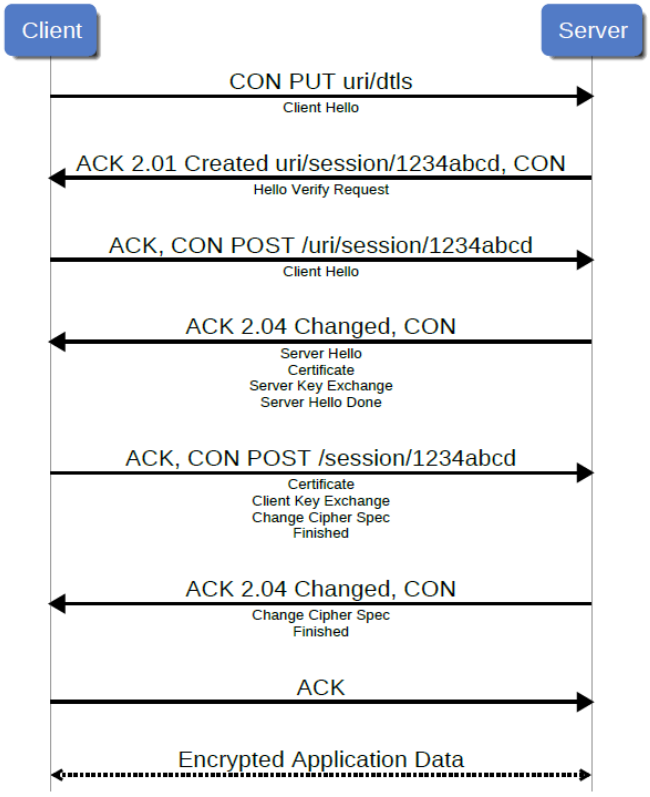

Figura 5: Acuerdo de llave DTLS sobre CoAP con llave pública (Raw Public Key).

Protocol	ROM	RAM
CoAP + Blip	51410 B	6653 B
standard DTLS	10983 B	7380 B
DTLS over CoAP	8936 B	7144 B

Figura 6: Resultados de la implementación de DTLS vs su versión estándar[10]. La fila CoAP + Blip corresponde a la pila generada por la capa MAC 802.15.4, el protocolo de ruteo RPL, 6LowPAN y CoAP.

4. Herramienta para IoT

Como se estudió en las secciones anteriores, existen ya protocolos para redes de bajo poder y consumo, pero es necesario poder programar y configurar estas redes, además que es esperable que muchas de estas tengan una gran cantidad de dispositivos lo que puede hacer muy compleja su depuración. El sistema operativo Contiki puede ayudar en esta tarea.

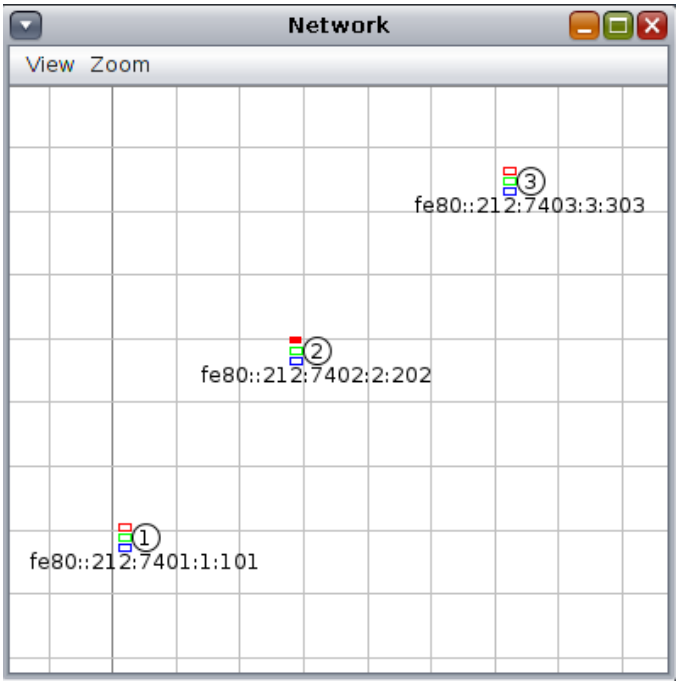

Figura 7: Red IoT ejemplo usando Cooja, 1 router de borde usando RPl y 2 motes con implementación de CoAP Erbium REST.

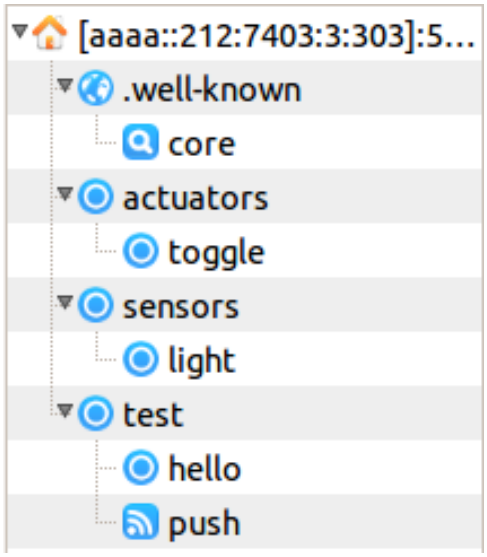

Figura 8: Estructura del Nodo 3 de la red, este se simuló usando un Sky Mote que implementa Erbium REST.

Contiki[13] es un sistema operativo diseñado para funcionar en una gran variedad de dispositivos de escasos recursos y conectarlos a Internet. Tiene además una versión llamada Instant Contiki basada en Ubuntu que tiene herramientas para configurar dispositivos e incluso simular redes de Nodos IoT.

Lamentablemente no se pudo implementar un sistema que utilizara todo lo estudiado en este proyecto, ya que es una herramienta muy compleja y que requiere de mucho más estudio para utilizar todas sus capacidades. Pero se pudo simular una red de nodos ejemplo para ver que efectivamente se puede aprovechar esta herramienta para la configuración de redes IoT.

Utilizando la aplicación Cooja se puede generar un mapa de nodos cada uno programado de la manera que se estime conveniente y estudiar la comunicación que existe entre ellos, también es posible simular distintos tipos de nodo dependientes del hardware de cada uno. Así se puede ver los resultados que tendrá la red previa a su implementación real en hardware, de esta manera esta herramienta se convierte en un instrumento muy poderoso si se quiere por ejemplo instalar una red de sensores con nodos IoT conectados a la red.

En este caso se implemento un ejemplo simple que utiliza un router de borde IPv6 mediante el protocolo RPL, y 2 nodos con la implementación de CoAP Erbium (er) REST el cual viene integrado para Contiki. De esta forma se puede acceder mediante una URI a los motes (por ejemplo coap://[aaaa::242:7402:2:202], la cual corresponde a uno de los motes con implementación de CoAP). De esta forma se tiene la red de la Fig. 7. Luego en FireFox con la extensión Copper (Cu) la cual permite acceder a direcciones con URI del tipo CoAP se puede acceder a una interfaz del mismo sensor, en donde se pueden hacer requerimientos del tipo GET, PUT, POST o hacer pings desde el Nodo 1 que es el router de borde.

Para poder realizar un ping o hacer un requerimiento del tipo POST, primero el nodo 1 tiene que hacer un GET de forma de poder conocer la estructura del nodo y que tipo de comandos enviar, la Fig. 8 muestra un ejemplo de la estructura del nodo 3 vista desde el browser. Luego de esto se puede por ejemplo enviar un 1 en la sección toggle del nodo, esto lo que hace en este caso es encender el led rojo, que se puede ver en la Fig. 7, esto gracias a las características del mote Sky. Luego se realizó un ping del nodo 1 al nodo 3, en la Fig. 9 se pueden ver los paquetes que corresponden a esta comunicación, primero se envía un paquete de 1 a 2, ya que es el único a su alcance, 2 envía un paquete a 1 y 3 que en su caso ambos estan a su alcance, así 3 le responde a 2, finalmente 2 envía el paquete a 1 y 3, terminando el ping en 1.

Finalmente es posible implementar DTLS en Cooja, pero para esto se requiere compilar librerías de DTLS para su uso en Contiki, existe una implementación hecha por Vladislav Perelman en [14]. Lamentablemente el agregar esto a la simulación no era una tarea sencilla, y no se tuvo el tiempo para hacerlo, pero nos demuestra que es posible agregar seguridad al protocolo CoAP.

Lo que se realizó en esta sección fue una simulación utilizando modelos de hardware real utilizando 3 Tmote Sky virtuales. Gracias a las capacidades del SO es posible aplicar esto en hardware real, pero lamentablemente no se tenía acceso a estos dispositivos.

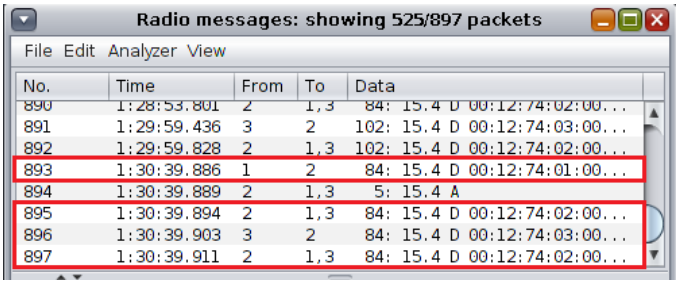

Figura 9: Paquetes correspondientes a un ping desde Nodo 1 a Nodo 3.

5. Conclusión

La Internet de las Cosas, o IoT, es un campo emergente que crecerá exponencialmente en los próximos años, ya lo podemos ver en la implementación de estas tecnologías actualmente. Por esta razón es fundamental estudiar la aplicación de estándares que permitan la comunicación entre dispositivos de gran variedad de características, capacidades y estructuras.

En este proyecto se estudió uno de los protocolos más utilizados en este ámbito, CoAP o Constrained Application Protocol, un protocolo de capa de aplicación para dispositivos de capacidades restringidas. Este protocolo tiene funcionalidades similares a HTTP y se puede traducir fácilmente a este, con la ventaja de poder ser utilizado en dispositivos IoT.

También se estudió la posibilidad de implementar seguridad sobre CoAP con DTLS, protocolo de capa de transporte basado en TLS sobre UDP. Lamentablemente este protocolo no se puede implementar directamente ya que es demasiado exigente en recursos para poder ser utilizado en dispositivos IoT, es por esto que diversos autores han propuesto soluciones para su implementación y existen librerías para hacerlo como TinyDTLS[15].

Una de las herramientas más útiles que se encontró en la investigación es Instant Contiki, un sistema operativo basado en Ubuntu con el cual se pueden simular redes IoT con la aplicación Cooja o también implementar en dispositivos reales el sistema operativo Contiki, diseñado para su uso en dispositivos de capacidades restringidas. Con el se pudieron realizar varias pruebas y utilizar el protocolo CoAP, el cual realmente es bastante similar en funcionamiento a HTTP.

Debido a lo complejo que es el sistema no se pudo implementar alguna versión de DTLS en las simulaciones para ver como funcionaba este. La herramienta tiene muchas funciones, aplicaciones e implementaciones de protocolos, lo que llevaría mucho tiempo poder tener el conocimiento adecuado para agregar funcionalidades nuevas.

Finalmente se puede decir que IoT es un campo que en los próximos años seguirá creciendo y nuevas soluciones e ideas irán apareciendo, por lo que involucrarse en este puede traer nuevos proyectos bastante interesantes y herramientas como estas pueden ser útiles en su desarrollo.

Instrucciones para hacer funcionar la red de ejemplo en Cooja

Para hacer funcionar la red de ejemplo que se mostró en la sección 4 primero se deben seguir los siguientes pasos:

- Descargar Instant Contiki 3.0 de [13] y montarlo en el programa para máquinas virtuales de preferencia, como VMWare (En la página aparece más información sobre el montaje).
- Iniciar la máquina virtual y ingresar como contraseña `user`.
- Abrir una terminal y ingresar el comando `cd contiki/tools/cooja` luego ingresar `ant run`.
- Se va a abrir la aplicación Cooja, en ella abrir el archivo enviado con el proyecto, llamado `cooja.csc`.
- En Cooja iniciar la simulación apretando el botón Start.
- Una vez este corriendo la simulación abrir otra terminal e ir a la carpeta `cd contiki/examples/ipv6/rpl-border-router` y luego ingresar `make connect-router-cooja`.
- Finalmente se puede abrir un browser que admita direcciones del tipo `coap://`, FireFox que viene incluido trae la extensión Copper (Cu) que permite hacerlo, dentro de este se puede abrir las direcciones `coap://[aaaa::212:7402:2:202]` o `coap://[aaaa::212:7403:3:303]`, para el nodo 2 y 3 respectivamente.

Teniendo la red configurada y funcionando, ahora se puede realizar pings a los nodos desde el browser, prender algún LED utilizando el método POST u otras cosas para ver un funcionamiento simple de una red IoT simulada con motes reales, en este caso el Sky mote.

Referencias

[1] "That 'internet of things' thing - page 1 - RFID journal," http://www.rfidjournal.com/articles/view?4986, 2016.

[2] A. Nordrum, "Popular internet of things forecast of 50 billion devices by 2020 is outdated," http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated, 2016.

[3] L. Columbus, "Internet of things on pace to replace mobile phones as most connected device in 2018," http://www.forbes.com/sites/louiscolumbus/2016/07/09/internet-of-things-on-pace-to-replace-mobile-phones-as-most-connected-device-in-2018/#4042597e6aef, 2016.

[4] Iot-a.eu, "Internet of things - architecture — iot-a: Internet of things architecture," http://www.iot-a.eu/public, 2016.

[5] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," Internet Requests for Comments, RFC Editor, RFC 7252, June 2014, http://www.rfc-editor.org/rfc/rfc7252.txt. [Online]. Available: http://www.rfc-editor.org/rfc/rfc7252.txt

[6] https://github.com/eclipse/californium, 2016.

[7] https://github.com/smeshlink/CoAP.NET, 2016.

[8] https://github.com/nning/coap, 2016.

[9] R. A. Rahman and B. Shah, "Security analysis of iot protocols: A focus in coap," in *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)*. IEEE, 2016, pp. 1–7.

[10] A. Capossele, V. Cervo, G. De Cicco, and C. Petrioli, "Security as a coap resource: an optimized dtls implementation for the iot," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 549–554.

[11] J. Park and N. Kang, "Lightweight secure communication for coap-enabled internet of things using delegated dtls handshake," in *2014 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2014, pp. 28–33.

[12] A. Sumits, "The history and future of internet traffic," http://blogs.cisco.com/sp/the-history-and-future-of-internet-traffic, 2016.

[13] "Contiki: The Open Source OS for the Internet of Things." [Online]. Available: http://contiki-os.org/

[14] https://github.com/renzoe/dtls-contiki/wiki, 2016.

[15] https://sourceforge.net/projects/tinydtls, 2016.