

# Performance Evaluation of HTTP/2 Window Size in the Internet of Things

Diego Londoño<sup>1,2</sup>, Maite González<sup>1,3</sup>, Sandra Céspedes<sup>1,2</sup>,  
Javier Bustos<sup>1,3</sup>, Gabriel Montenegro<sup>4</sup>

NIC Chile Research Labs<sup>1</sup>

Departamento de Ingeniería Eléctrica, Universidad de Chile<sup>2</sup>

Departamento de Ciencias de La Computación, Universidad de Chile<sup>3</sup>

Microsoft<sup>4</sup>

[diegolondono, maite]@niclabs.cl

## Resumen

Constrained devices are a common factor in the Internet of Things (IoT). These devices have limited RAM and ROM memory, reduced battery, processing capacity, and transmission power. In consequence, these devices may not work properly with traditional Internet protocols like TCP and HTTP, which were not created for constrained scenarios. However, in 2015 the Internet Engineering Task Force (IETF) published the newest version of the most popular application protocol on the Internet: HTTP/2. It has significant improvements over the previous version, such as binary frames, multiplexing of streams, priorities, flow control, among others. In this work, the parameter of HTTP/2 window size is evaluated in terms of use of CPU, use of memory, response times, and energy consumption for constrained devices in IoT. With this work, we expect to promote the discussion about the utilization of HTTP/2 for IoT and to contribute to an eventual standardization.

Key index: Application Layer Protocol, Constrained devices, HTTP/2, IETF.

## 1. Introducción

*Hypertext Transfer Protocol* (HTTP) es un protocolo de capa de aplicación que permite la transferencia de información a través de la Web. Aunque el protocolo existe desde principios de los años noventa, la versión HTTP/1.1, publicada en 1999, ha sido la más difundida. Con el paso de los años esta versión empezó a evidenciar algunas falencias, tales como cabeceras repetitivas, necesidad de crear una conexión TCP

por cada solicitud, la utilización de una estructura ASCII, entre otras. Lo anterior llevó a la publicación de HTTP/2 [Bel15], en 2015, que no sólo suple las carencias de la versión previa del protocolo, sino que agrega nuevas funcionalidades con el fin de agilizar la transferencia de información en las aplicaciones web actuales. Entre las principales características de HTTP/2 están: La estructura binaria del protocolo, compresión de cabeceras, multiplexión de *streams*, control de flujo, establecimiento de prioridades, y *Server Push*, una nueva funcionalidad que permite enviar anticipadamente recursos al cliente, ahorrando así tiempo entre peticiones.

Actualmente, según la *IoT Developer Survey 2017*, HTTP/1.1 es el protocolo de aplicación más utilizado en *Internet of Things* (IoT) [IoT17], campo donde son típicas las comunicaciones *Machine to Machine* (M2M) entre dispositivos restringidos (con limitada capacidad de procesamiento, poca memoria y/o alimentados por baterías). Ante tales limitaciones, distintas entidades han hecho esfuerzos para crear protocolos livianos que se adapten a ese tipo de ambientes restringidos. Es así como fueron creados los protocolos de aplicación para *IoT: Constrained Application Protocol (CoAP)* y *Message Queue Telemetry Transport (MQTT)*. A pesar de los esfuerzos de creación de protocolos adecuados a las necesidades, HTTP sigue siendo el protocolo de comunicación más utilizado en el campo, por lo que se ha llevado a proponer una adaptación de HTTP/2 para IoT.

Dado lo anterior, la propuesta en [Mon17] señala que a pesar de que HTTP/2 no fue creado para utilizarse en dispositivos IoT, es compacto, flexible y configurable, por lo que variando algunos valores en los parámetros de configuración se lograría un funcionamiento adecuado del protocolo en condiciones restringidas. El *Internet draft* no define valores exactos, por lo cual se requiere una evaluación de desempeño que lleve a la definición de los valores que tendrían los parámetros en una versión adaptada a IoT.

En este trabajo se evalúa el desempeño de HTTP/2 al variar el tamaño de la ventana de control de flujo, en términos de uso de CPU, uso de memoria, consumo energético y tiempos de respuesta, con miras a encontrar los primeros valores que servirían para adaptar HTTP/2 a un entorno restringido. Cabe destacar que aunque HTTP/2 no define un algoritmo específico para el control de flujo, el funcionamiento respecto a las ventanas deslizantes es similar al de TCP. La contribución de este trabajo radica en describir el comportamiento del protocolo en un escenario IoT e ilustrar los valores para los cuales se usan mejor los recursos restringidos de los dispositivos empleados en el experimento. La implementación de HTTP/2 utilizada es Nghttp/2 [Ngh17] y se corre sobre una Raspberry Pi 3 Modelo B.

## 2. Metodología

En el perfil de configuración propuesto en [Mon17], se identifican los parámetros que deberían cambiar sus valores para un correcto funcionamiento del protocolo en entornos de IoT. En este trabajo el tamaño de la ventana inicial (SETTINGS\_INITIAL\_WINDOW\_SIZE) es variado y evaluado en términos de uso de CPU, tiempos de respuesta y consumo energético.

Nghttp2, una implementación cliente/servidor de HTTP/2 desarrollada en C, ha sido utilizada en este trabajo. Esta aplicación se corrió sobre dos Raspberry Pi 3 modelo B, una para el cliente y otra para el servidor. Las dos Raspberry se conectaron a través de un *switch* utilizando *FastEthernet*, formando así una red local. El objetivo de usar un medio cableado es concentrar la evaluación en los dispositivos restringidos y no en el desempeño de una posible red inalámbrica. Por otro lado, el consumo energético del cliente se mide utilizando un módulo ACS712 y una Arduino Mega. Este módulo permite medir la corriente del cable que suple energéticamente a la Raspberry Pi. Asumiendo que el voltaje de alimentación se mantiene en 5 V, la potencia fue calculada como el producto de la corriente y el voltaje. Con los datos de la potencia instantánea y la duración de cada proceso, se calculó el trabajo requerido para cada caso, tomando como base que la potencia instantánea utilizada por la Raspberry en estado de reposo es de 80 mW 0 a 900 mW.

El servidor web a consultar aloja una página web con un index html que contiene un archivo css, dos archivos javascript y cuatro imágenes jpg. Los archivos en conjunto suman alrededor de 7 MB. La metodología se basa en un cliente que realiza una serie de peticiones al servidor (68 en total), mientras se miden datos de uso de CPU, uso de memoria y tiempo de CPU tanto del lado del cliente como del servidor. Además se mide el tiempo de respuesta real y el consumo energético del cliente variando en cada iteración el parámetro evaluado. En el caso del tamaño de la ventana, se evaluó la

ventana desde 256 hasta 65535 (el valor por defecto en HTTP/2).

Considerando la fórmula para calcular el tamaño de la muestra en una población infinita, los resultados obtenidos tienen un 90 % de confianza, aunque hay que resaltar que los resultados dependen en buena medida de la implementación de HTTP/2 utilizada. Otras implementaciones con mejor o peor desempeño podrían arrojar valores resultantes diferentes, aunque el comportamiento general debería ser similar al del presente trabajo.

## 3. Resultados

Los resultados de las pruebas experimentales dejan ver el comportamiento de las métricas en el cliente y servidor. En la figura 1 se muestra el comportamiento de los tiempos de respuesta, tiempo de CPU y el trabajo requerido por el cliente ante la variación del tamaño de la ventana. Se puede ver que cuando la ventana es muy pequeña (256 Bytes), el largo tiempo en el que el servidor recibe paquetes provocan un mayor uso de la CPU, y esto se traduce en un mayor trabajo. En la medida que el tamaño de la ventana aumenta, los tiempos de respuesta disminuyen al igual que el tiempo de CPU. A partir de 4096 Bytes como tamaño de ventana, el trabajo requerido se estabiliza alrededor de 120 mJ.

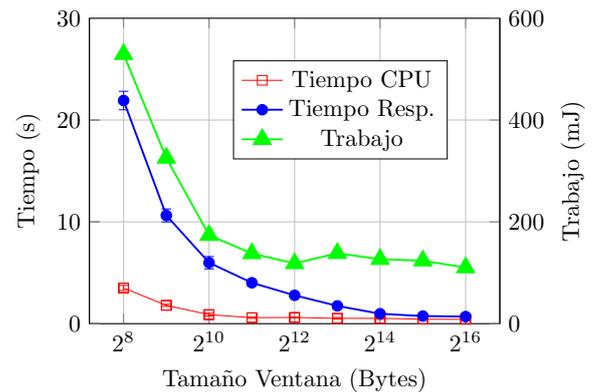


Figura 1: Tiempo de Respuesta, Tiempo de CPU y Trabajo versus tamaño de ventana en el cliente.

En la figura 2 se muestran los datos obtenidos en el servidor. Los tiempos de respuesta y tiempos de uso de CPU tienen un comportamiento similar al del cliente. Ambas métricas tienen una tendencia decreciente a medida que aumenta el tamaño de la ventana, hasta llegar a 427 ms en el tiempo de CPU y 627 ms en tiempo de respuesta cuando la ventana es 65535 Bytes.

En la figura 3 se muestra el porcentaje de uso de CPU tanto en cliente como en el servidor ante la variación de la ventana. La gráfica muestra un comportamiento similar en las dos partes, aunque en diferentes magnitudes. Hay una relativa estabilidad hasta 4096 Bytes, y después de esto el uso aumenta significativa-

mente, siendo mayor en el cliente, donde llega a ocupar el 60% de capacidad de CPU cuando la ventana es 65.535 Bytes, mientras que en el servidor es aproximadamente de un 26%.

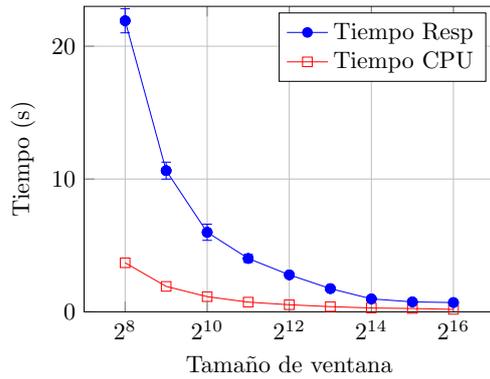


Figura 2: Tiempo de Respuesta y Tiempo de CPU versus tamaño de ventana en el servidor

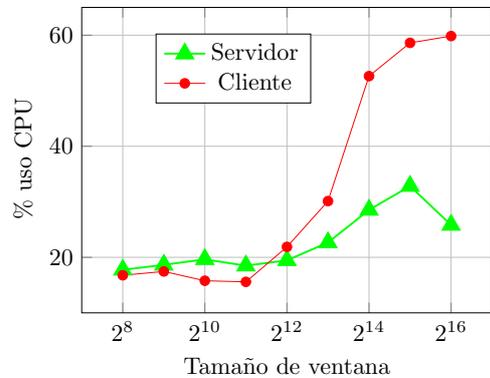


Figura 3: Porcentaje CPU versus tamaño de ventana en cliente y servidor

De las gráficas 1 y 3 se puede observar que el trabajo realizado por el cliente y el porcentaje de uso de CPU en el servidor y cliente tienen un punto de quiebre alrededor de 4096 Bytes. En ese punto, según los datos del cliente, el trabajo es muy similar al que hace el dispositivo en el valor por defecto del protocolo (65535 Bytes), sin embargo, el tiempo de CPU aumenta un 42.17% y los tiempos de respuesta aumentan un 299.5%.

## 4. Conclusiones

El tamaño de la ventana tiene un impacto significativo en el desempeño del protocolo HTTP/2 en entornos IoT. A medida que aumenta su tamaño, los tiempos de respuesta y de CPU se acortan, mientras que la potencia instantánea consumida aumenta; sin embargo, al realizarse los procesos de manera más rápida, el trabajo es menor que cuando la ventana es pequeña. Este comportamiento se evidencia tanto en el cliente como en el servidor.

Según la información recopilada en este estudio, en términos energéticos el valor sugerido para el tamaño de la ventana es 4096 Bytes. Este tamaño puede cambiar dinámicamente según las condiciones de la comunicación. De esta manera cuando un dispositivo tiene una alta carga de trabajo, no tiene suficiente capacidad en cuanto a *hardware* o el canal tiene muchas pérdidas, el tamaño de la ventana podría reducirse. Si por el contrario las condiciones son buenas, el tamaño de la ventana podría aumentar.

En este estudio, la variación del número de *streams* concurrentes también fue evaluada, sin embargo, no se observó una tendencia clara en cuanto a las métricas utilizadas. Esto puede deberse a la forma como está constituida la página web. Corresponde a trabajos futuros hacer las pruebas necesarias que permitan evidenciar mejor el efecto de la cantidad de *streams* concurrentes, así como el impacto de redes inalámbricas con pérdidas en el desempeño del protocolo.

## 5. Trabajo Futuro

En futuros trabajos realizaremos una evaluación de los demás parámetros mencionados en el *Internet draft* del perfil de HTTP/2 para IoT [Mon17]. De igual manera se harán pruebas similares en dispositivos tipo 2 y 1 según la clasificación de dispositivos restringidos que hace el RFC 7228 [Ker14]. En particular se debe evaluar el desempeño ante la variación de la cantidad de *streams* con una página web que tenga recursos de tamaños similares cuando los objetos a transmitir son similares en carga.

## 6. Acknowledgements

Este trabajo es parcialmente financiado por el proyecto FONDECYT Iniciación 11140045 Instituto de Sistemas Complejos de Ingeniería (CONICYT: FB0816).

## Referencias

- [Ker14] C. Bormann, M. Ersue, and A. Keranen, *Terminology for Constrained-Node Networks*, RFC 7228, May 2014. Disponible: <http://www.rfc-editor.org/info/rfc7228>.
- [Bel15] M. Belshe, R. Peon, and M. Thomson, Ed., *Hyper-text Transfer Protocol Version 2 (HTTP/2)*, RFC 7540, May 2015. Disponible: <https://tools.ietf.org/html/rfc7540>.
- [Mon17] G. Montenegro, S. Cespedes, S. Loreto and R. Simpson., *HTTP/2 Configuration Profile for the Internet of Things*, Mar 2017. Disponible: <https://github.com/h2otwg/h2ot-profile/blob/master/draft-montenegro-httpbis-h2ot-profile-00.txt>.
- [IoT17] IoT Eclipse, IEEE Internet of Things, *IoT Developer Survey 2017*, April 2017. Disponible: <https://www.slideshare.net/IanSkerrett/iot-developer-survey-2017>.
- [Ngh17] Nghttp2, HTTP/2 C library and Tools. [Online]. Disponible en: <https://github.com/nghttp2/nghttp2>.