Web Protocols and Challenges of Web Latency in the Web of Things

Nitin Naik and Paul Jenkins

Defence School of Communications and Information Systems
Ministry of Defence, UK
Email: nitin.naik100@mod.uk and paul.jenkins683@mod.uk

Abstract—**The Internet of Things (IoT) has been substantially dominated by proprietary and domain specific protocol stacks. There is no universal application protocol for the IoT that can work across many networking interfaces available today. The successful implementation of the IoT requires a single universal application layer protocol for devices and applications to talk to each other, regardless of how they are physically connected. One of the simplest and apparent solutions is to reuse mechanism, which is already extensively used for building scalable and interactive applications, such as the World Wide Web (Web) itself. Therefore, the adoption of the Web ecosystem and infrastructure to build applications for the IoT, leads to the concept of the Web of Things (WoT) and extends the IoT with the amalgamation of the Web as an open IoT ecosystem based upon open standards. While the IoT has been focusing on lower layers and hardware infrastructure, the WoT relies exclusively on application level protocols and tools. Web protocols are a critical factor in the successful implementation of the WoT. However, one of the main issues is web latency that may significantly affect the real-time performance of IoT systems. Therefore, this paper conducts a number of practical investigations on the performance and web latency of application layer protocols: HTTP/1.1, SPDY and HTTP/2. Using experimental results, it analyses the challenges of web protocols for the implementation of WoT.**

Keywords—Internet of Things, Web of Things, Web Latency, HTTP, SPDY, HTTP/2

I. INTRODUCTION

The Internet-of-Things (IoT) assumes a future for the Internet where digital and physical entities can be linked, using appropriate information and communication technologies, to enable a whole new class of applications and services [1]. The IoT has been growing rapidly and several researchers have predicted that trillions of "smart things" will connect the physical environment to the Internet in the near future. One of the biggest obstacles to the expansion of IoT is its proprietary and domain specific projects that lead to hundreds of incompatible protocols [2]. This makes the integration of data and services from various devices extremely complex and costly. To enable this vision of an open IoT, a single universal application layer protocol is required for devices and applications to talk to each other, regardless of how they are physically connected. This issue can be solved by re-using the existing World Wide Web (Web) infrastructure. Therefore, the adoption of the Web ecosystem and infrastructure to build applications for the IoT gives rise to the concept of the Web of Things (WoT) [3]. In the WoT, any device can be accessed using standard Web protocols. Thus, WoT can connect the sensor world and the Web world together and extends the IoT with the amalgamation of the Web as an open IoT ecosystem based upon open standards [2].

The key component of the Web is protocols and thus for the WoT. Adopting the Web infrastructure and web protocols for WoT is a complex task as the IoT requires interaction between smart objects rather than human beings. Regarding this, the web protocols have the biggest challenge of reducing the web latency to make them become suitable for the IoT and WoT. Therefore, this paper conducts the practical investigations on the performance and web latency of application layer protocols: HTTP/1.1, SPDY and HTTP/2. It has carried out four different experiments to examine the use of SPDY and HTTP/2 in reducing the web latency and suitability for the WoT. Thsee experimental results have suggested that the implementation of this protocols at the server may perhaps improve the performance and reduce the web latency, however, the client support did not find any great improvement in the performance. Indeed, web protocols are one of the major factors responsible for web latency; however, the wide variation (based on these experiments) in load times for similar types of websites suggest a heavy dependency of web latency on the contents of the website and the location of the server. These factors were not considered in this investigation. Finally, based on the experimental results it analyses the challenges of web protocols for the implementation of WoT.

The remainder of this paper is organised as follows: Section II introduces the theoretical background of Internet of Things, Web of Things, web latency, HTTP/1.1, SPDY and HTTP/2; Section III illustrates the practical investigations of web latency in HTTP/1.1, SPDY and HTTP/2 web protocols; Section IV elucidates the feasibility study of SPDY and HTTP/2 for the use in Web of Things; Finally, Section V concludes the paper and suggests some future areas of extension.

II. THEORETICAL BACKGROUND

This section presents the theoretical background of Internet of Things, Web of Things, web latency, HTTP/1.1, SPDY and HTTP/2.

A. Internet of Things (IoT)

A number of research studies have envisioned that trillions of "smart things" will connect the physical environment to the Internet in the near future. A "smart thing" can be defined as any physical device or object with the capabilities of sensing,

actuating, computation and communication. This phenomenon is commonly viewed as the Internet of Things (IoT) where smart objects carry out various activities on the Internet. It is not necessary that the thing must be physically connected to the Internet. The communication network used can be an Auto-ID technology, short-range radio (e.g., Bluetooth, ZigBee) or a local Wi-Fi network. However, in its present form IoT is often described as a collection of isolated Intranets of Things that cannot really interact with each other. Therefore, building a single global ecosystem of Things communicating with each other seamlessly is virtually impossible today.

B. Web of Things (WoT) and its Need for IoT

The IoT has been largely dominated by proprietary and domain specific protocol stacks. There is no universal application protocol for the Internet of Things that can work across the many networking interfaces available today. For making the IoT a reality, there will be a need for a single universal application layer protocol for devices and applications to talk to each other, regardless of how they are physically connected. The most striking idea is to reuse something that is already widely adopted for building scalable and interactive applications, such as the Web itself? Therefore, the next logical step is to use the Web ecosystem and infrastructure for building IoT-based applications, overcoming this ongoing "one device, one protocol, one app" pattern [3]. This gives rise to the concept of the Web of Things (WoT) and extends the IoT with the amalgamation of the Web as an open IoT ecosystem based upon open standards.

The main aim of the WoT is to standardize the use of existing web technologies to facilitate the building of applications and web services for the IoT. The WoT reuses and leverages readily available and widely accepted web protocols, web standards and blueprints to make data and services offered by smart objects more accessible to every web user and developer. This explores the possibility of using scripting languages such as JavaScript, data encodings such as JSON and EXI, and protocols such as HTTP and WebSockets. The implementation of the WoT concept will significantly expand the access and effect of the current Internet of Things by achieving uniform functionality among smart objects. The Web of Things only concerns with the top OSI Application Layer, which handles applications, services and data. In contrast, IoT is concerned with the lower levels of the OSI stack related to the hardware devices, physical layers, and connections.

C. Web Latency

Web latency is the time it takes for the web server to receive and process a request for a page object from the web client. It depends on a number of factors, such as web protocols and web applications; however, it largely depends on how far away the client is from the server. Here, the main focus of this investigation is on web protocols such as HTTP. HTTP is an application-layer protocol providing basic request/response semantics for transporting content over the web (see Fig. 1). TCP is a reliable transport layer protocol supporting HTTP for all the underlying services on the web such as for guaranteed delivery, duplicate suppression, in-order delivery, flow control and congestion avoidance [4]. Both are responsible for web latency; unfortunately, neither protocol
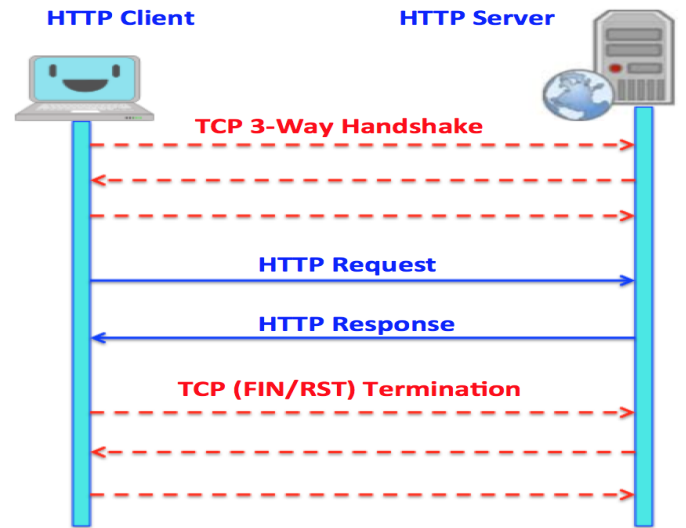

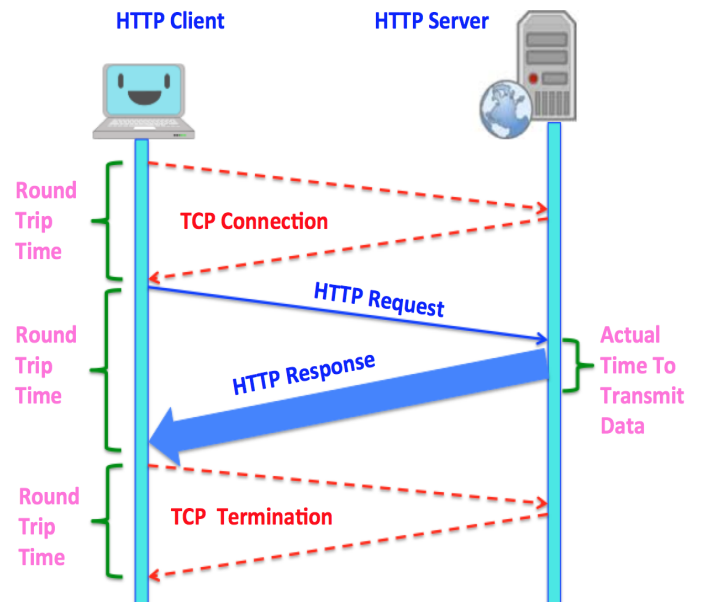Fig. 1. HTTP Client-Server Communication

Fig. 2. HTTP Round Trip Time

was particularly designed with latency in mind. One of the most significant issues with HTTP is that it incurs many more round trips than necessary to retrieve the web objects (see Fig. 2). Furthermore, TCP is not helpful in reducing web latency because of its robustness, with its most remarkable feature being the three-way handshake needed to open a connection for every HTTP round trip. Additionally, TCP does not fully utilize the available network bandwidth for the first few round trips of a connection because of its "slow start technique", which is used to avoid network congestion. Consequently, the current web protocols especially HTTP-TCP are a major bottleneck of web latency. This problem may be one of the biggest impediments in adopting the web protocols in the implementation of WoT for an open IoT ecosystem based upon open standards.

D. Web Protocols

As mentioned earlier, web protocols are crucial in web latency. This subsection will discuss the web latency issue in the present HTTP/1.1 communication protocol and the emergence of the two new alternative protocols SPDY and HTTP/2 for reducing web latency.
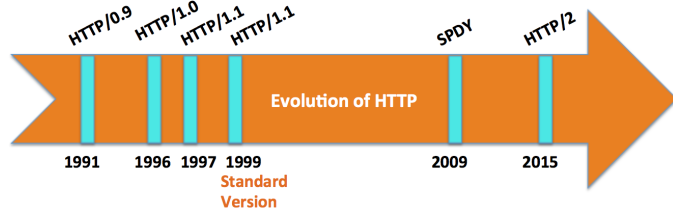

Fig. 3.   Evolution of Web protocol HTTP

1) HTTP/1.1 Protocol: HTTP/1.1 was first published by the IETF in 1997 and it has worked successfully for around two decades. The HTTP 1.1 Working Group has improved the performance and reduced the latency of HTTP/1.1 with the introduction of persistent connections (see Fig. 4), request pipelining (see Fig. 5), and chunked transfer encoding [5]. However, these features of HTTP/1.1 such as request pipelining have effectively failed due to the lack of support and deployment challenges; while some browsers today support pipelining as an optional feature, which forces strict request queuing on the client [5].

2) SPDY Protocol: SPDY is an application-layer protocol for transporting content over the web. It is an experimental protocol, developed by Google in 2009. The main aim of designing SPDY was to minimise web latency by up to 50%[4]. It does not replace HTTP protocol but it augments it by adding a number of features that assist in making web transactions faster. Akin to HTTP, SPDY also uses TCP as the underlying transport layer; therefore, it does not demand any change in the
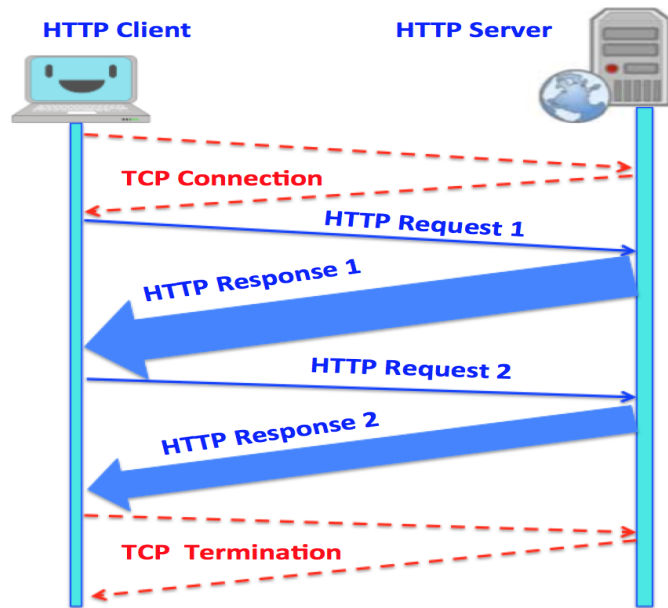

Fig. 4.   HTTP/1.1 persistent connection for improving performance
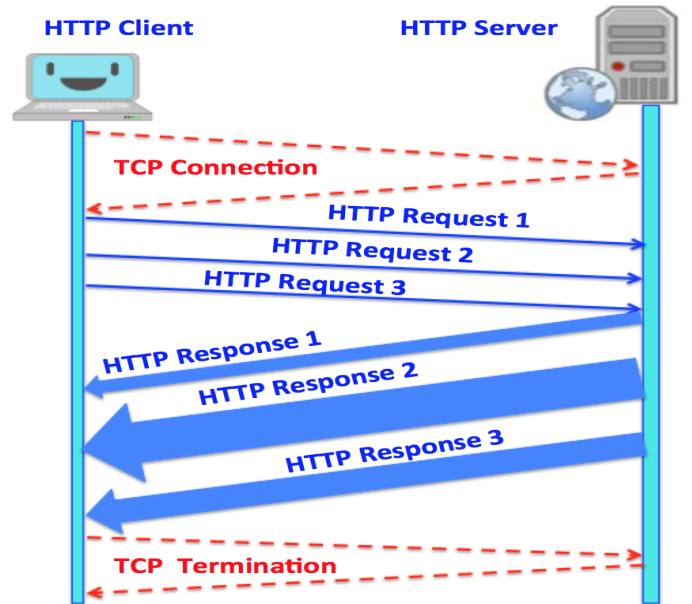

Fig. 5.   HTTP/1.1 pipelined connection for improving performance

existing networking infrastructure. The practical requirement of SPDY is the use of TLS/SSL, but it is not compulsory. Therefore, the end-to-end encrypted TLS/SSL tunnel allows the client and the server to exchange SPDY frames without the intervention of intermediate nodes. Thus, how does the client and server know to use SPDY once the TLS/SSL tunnel is opened? For this, a new protocol Next Protocol Negotiation (NPN) is used. NPN is a TLS/SSL extension, which allows the client and the server to negotiate the application protocol as part of the TLS/SSL handshake. It also eliminates the extra round trip to negotiate the application protocol. This is one of its main advantages as compared to the current WebSocket handshake, which imposes another round trip of latency in addition to the SSL negotiation. The development of SPDY had been developed to the version SPDY3.x with SPDY4 not being released as a separate specification but becoming an alias for the new HTTP/2 standard. In HTTP/2, NPN has now been deprecated. SPDY's design requirement of TLS/SSL limited its actual adoption.

3) HTTP/2 Protocol: HTTP/2 was recently introduced in 2015, nearly two decades after its predecessor HTTP/1.1. HTTP/2 has been developed by the IETF HTTP Working Group, however, it is mainly based on Google's experimental SPDY protocol. It enables a more efficient use of network and server resources, and a reduced perception of latency by introducing header field compression facility and allowing multiple concurrent exchanges on the single connection from browsers to a Web site. HTTP/2 also introduces unsolicited push of representations from servers to clients. HTTP/2 replaces HTTP/1.1 on the wire only but maintains the HTTP/1.1 message syntax. Therefore, all HTTP methods, status codes and semantics are the same, and it is possible to use the same APIs as HTTP/1.1 with some alterations to represent the new version [6], [7]. Similar to SPDY NPN protocol, HTTP/2 employs TLS/SSL extension protocol, called Application-Layer Protocol Negotiation (ALPN) within the TLS/SSL handshake. In this case multiple application protocols are supported on

the same TCP or UDP port. Furthermore, it also allows the application layer to negotiate which application protocol will be used within the TLS/SSL connection.

III. PRACTICAL INVESTIGATIONS OF WEB LATENCY IN HTTP/1.1, SPDY AND HTTP/2

Web latency can be improved at the application and transport layers. However, this investigation is only focused on the application layer protocols HTTP/1.1, SPDY and HTTP/2. Various research studies have been conducted on the performance of SPDY (because HTTP/2 was only introduced a few months ago), some have confirmed that SPDY has reduced the web latency [8], [9], however, some have even presented the opposite effect [10]. The later studies are crucial as SPDY forms the basis of new HTTP/2 protocol. This practical investigation covers both SPDY and HTTP/2 protocols including the existing HTTP/1.1. Here, four different experiments are conducted to investigate the performance and web latency of SPDY and HTTP/2 over on HTTP/1.1. The results of the investigation have been analysed for performance and web latency when client and server both support SPDY and HTTP/2 protocols, when only one supports and when both do not support. The two client browsers chosen are Google Chrome (a version that supports SPDY and HTTP/2) and Mozilla Firefox (version that does not support SPDY and HTTP/2). Similarly, to access the server resources, two types of websites are chosen one that supports SPDY and HTTP/2 and the other which does not support SPDY and HTTP/2. It should be noted this was the status of these browsers and websites at the time of the experiment, however, the whole process is in perpetual change and the current status may be completely different. The other investigation tools used in these experiments are HAR (HTTP Archive) Analyser, Wireshark and several websites to identify the current status of browsers and websites. This investigation has focused on three parameters: average load time, number of request and amount of bytes in for the first document view and repeat document view to gain insight from the user's point of view only. HTTP performance logs including these parameters are recorded in the HAR files. HAR (HTTP Archive) is a file format used by several http session tools to track all the logging of the web browser's interaction with a site.

A. Experiment-1: SPDY-HTTP/2-Excluded Website using SPDY-HTTP/2-Excluded Browser

In the first experiment, six random websites are chosen that do not support SPDY and HTTP/2 protocols at the time of experiment. They are probed through a version of Mozilla Firefox browser that does not support SPDY and HTTP/2. Therefore, this experiment is based on only the existing HTTP/1.1 protocol and to check its performance and web latency. To avoid errors and other side effects, this experiment is repeated 50 times and finally, the average value of each parameter is calculated as shown in Table I.

B. Experiment-2: SPDY-HTTP/2-Excluded Website using SPDY-HTTP/2-Enabled Browser

In the second experiment, the same six websites are chosen that do not support SPDY and HTTP/2 protocols at the time of experiment. However, they are probed through Google Chrome browser that supports SPDY and HTTP/2 protocols at the

TABLE I. PROBING SPDY-HTTP/2-EXCLUDED WEBSITE USING SPDY-HTTP/2-EXCLUDED BROWSER (MOZILLA FIREFOX VERSION THAT DOES NOT SUPPORT SPDY AND HTTP/2)

SPDY-HTTP/2-	First View			Repeat View		
Excluded Website	Time (S)	Request	Bytes (KB)	Time (S)	Request	Bytes (KB)
ieee.org	7.022	204	2028	5.023	66	367
one.com	5.646	57	1110	3.548	20	198
godaddy.com	2.246	13	361	1.573	2	96
names.co.uk	5.219	69	1468	4.051	31	104
bbc.co.uk	7.795	113	1328	5.44	19	148
skynews.com	5.326	72	1121	4.58	15	41

time of experiment. Therefore, this experiment is based on the partial support of SPDY and HTTP/2 and to check its performance and web latency. Similar to the first experiment, for avoiding errors and other side effects, this experiment is repeated 50 times and finally, the average value of each parameter is calculated as shown in Table II.

TABLE II. PROBING SPDY-HTTP/2-EXCLUDED WEBSITE USING SPDY-HTTP/2-ENABLED BROWSER (GOOGLE CHROME VERSION THAT SUPPORTS SPDY AND HTTP/2)

SPDY-HTTP/2-	First View			Repeat View		
Excluded Website	Time (S)	Request	Bytes (KB)	Time (S)	Request	Bytes (KB)
ieee.org	8.319	204	2030	6.598	75	381
one.com	4.126	55	1114	3.245	20	195
godaddy.com	2.821	13	360	1.881	2	96
names.co.uk	4.697	69	1468	3.182	31	105
bbc.co.uk	8.191	111	1319	5.173	17	142
skynews.com	5.873	77	1128	5.241	18	58

The first two experiments are conducted on the same websites that do not support SPDY and HTTP/2, however, one browser supports and other does not support SPDY and HTTP/2. One of the interesting findings of the comparative analysis of the results is that the client browser's support of SPDY and HTTP/2 does not affect the performance and web latency significantly. Additionally, this is also confirmed by the results where HTTP/1.1 supported browser has performed better than SPDY and HTTP/2 supported browser. Perhaps, this indicates the effects of other factors on the performance and web latency, which are not considered in this investigation. Finally, web protocols are one of the major factors responsible for web latency, however, the variation in load times for similar types of websites suggests that web latency is heavily dependent on the contents of the website and location of the server.

C. Experiment-3: SPDY-HTTP/2-Enabled Website using SPDY-HTTP/2-Excluded Browser

In the third experiment, six random websites are chosen that support SPDY and HTTP/2 protocols at the time of experiment. However, they are probed through a version of Mozilla Firefox browser that does not support SPDY and HTTP/2. Therefore, this experiment is based on the partial support of SPDY and HTTP/2 and to check its performance and web latency. Similar to the previous experiments, for avoiding errors and other side effects, this experiment is also repeated 50 times and finally, the average value of each parameter is calculated as shown in Table III.

SPDY-HTTP/2-	First View			Repeat View		
Enabled Website	Time (S)	Request	Bytes (KB)	Time (S)	Request	Bytes (KB)
facebook.com	2.501	36	368	1.966	2	35
yahoo.com	4.559	37	887	3.343	12	212
whatsonafrica.org	4.552	71	2217	2.261	2	58
twitter.com	6.339	58	1877	3.526	3	43
examsdocs.com	2.928	61	327	2.546	4	30
douban.com	9.177	79	1396	7.841	17	162

D. Experiment-4: SPDY-HTTP/2-Enabled Website using SPDY-HTTP/2-Enabled Browser

In the last experiment, the same six websites are chosen that support SPDY and HTTP/2 protocols at the time of experiment. They are probed through Google Chrome browser that also supports SPDY and HTTP/2 protocols. Therefore, this experiment is based on the full support for SPDY and HTTP/2 and to check its performance and web latency. Similar to the previous experiments, for avoiding errors and other side effects, this experiment is also repeated 50 times and finally, the average value of each parameter is calculated as shown in Table IV.

SPDY-HTTP/2-	First View			Repeat View		
Enabled Website	Time (S)	Request	Bytes (KB)	Time (S)	Request	Bytes (KB)
facebook.com	3.928	36	370	2.271	2	35
yahoo.com	3.962	39	880	3.353	11	203
whatsonafrica.org	4.706	71	2217	2.903	2	58
twitter.com	6.070	59	1909	3.459	4	50
examsdocs.com	3.601	61	333	2.918	4	30
douban.com	8.950	79	1328	7.290	18	174

The last two experiments are conducted on the same websites that support SPDY and HTTP/2, however, one browser supports and other does not support SPDY and HTTP/2. Both experimental results clearly suggest that the support of SPDY and HTTP/2 at the server (website) is crucial for the success of SPDY and HTTP/2 and could reduce the overall latency as compared to the websites that do not support SPDY and HTTP/2. However, comparative analysis of the results of the two different client browsers shows very minor changes in the performance and similar patterns as the first two experiments. Therefore they do not affect the performance and web latency greatly. This is also confirmed by the results where a HTTP/1.1 supported browser has performed better than SPDY and HTTP/2 supported browser. Again, this perhaps indicates the effects of other parameters on the performance and web latency, which was not considered in this investigation. As mentioned earlier, web protocols are one of the major factors responsible for web latency, however, the wide variation (based on this observation a more thorough evaluation is needed) in load times for similar types of websites suggests a heavy dependency on the contents of the website and location of the server. Overall, these two experimental results reveal that the support of SPDY and HTTP/2 at server could improve the performance and reduce the web latency to some extent.

IV. CHALLENGES OF WEB LATENCY IN WOT AND POTENTIALS OF WEB PROTOCOLS

Managing and coordinating real-time performance in the IoT will pose a host of new challenges. In the world of machine to machine interactions, everyday actions and as expectations become a more complex behind the scenes, therefore, every type of latency matters. The real-time IoT applications require a real-time response on the web. Even for those IoT applications that do not require instantaneous responsivity, they still must detect and react in close to real-time [11]. The data must be collected and processed continuously and with controlled latency. It emphasises the requirement for real-time applications in all the IoT situations with no place for batch processing models. Based on the web latency results obtained in the previous section, the minimum response time is around 2 seconds even using the enhanced web transport protocols SPDY and HTTP/2. This is a much higher response time for smart things and for their interaction in IoT. Additionally, if the smart device is further away from the point at which the IoT data is collected by the sensors, the longer it takes to get a response back to the device so that it can act upon [12]. However, distance-related network latency can be reduced by pushing data and processing closer to the IoT device where possible, but web applications will remain susceptible to poor routing decisions and network congestion [11].

The biggest concern of SPDY and HTTP/2 web protocols is the pace of acceptance in the web community itself. According to the W3techs.com website [13], [14], the usage statistics of SPDY and HTTP/2 on the web are 6.7% and 6.6% in February 2016 as shown in Figs. 6 and 7. Figs. 6 and 7 show the historical trend in the percentage of websites using SPDY and HTTP/2 up to February 2016. Perhaps, one of the biggest obstacles to the adoption of SPDY and HTTP/2 is the practical compulsion of TLS/SSL. However, SPDY is submerged into HTTP/2 and HTTP/2 which was only launched recently, therefore, it will take time for these protocols to be adopted by the majority of the web community. Simultaneously, WoT as an upper application layer of IoT is gaining momentum and acceptance, therefore, it is crucial to investigate the accepted WoT latency for HTTP/2 specifically in the case of IoT. Finally, based on the experiments conducted here and current recognition of HTTP/2 indicate that HTTP/2 requires review for WoT implementation, otherwise, replacement with more lightweight transport protocol such as MQTT, CoAP and AMQP to reduce the web latency.

V. CONCLUSION

This paper has presented a practical investigation of the performance and web latency of application layer protocols: HTTP1.1, SPDY and HTTP/2. These web protocols are the key protocols, which determine the performance of the Web and affect the web latency. Without an effective application layer protocol, the successful implementation of Web of Things (WoT) will not be possible. Consequently, this would be an impediment for making the open IoT ecosystem based upon open standards. In aspiring to achieve the vision of IoT, web latency is one of the main issues in using web protocols, therefore, this paper conducted several experiments to determine whether the new SPDY and HTTP/2 protocols reduce the web latency as compared to the existing HTTP/1.1 protocols. The
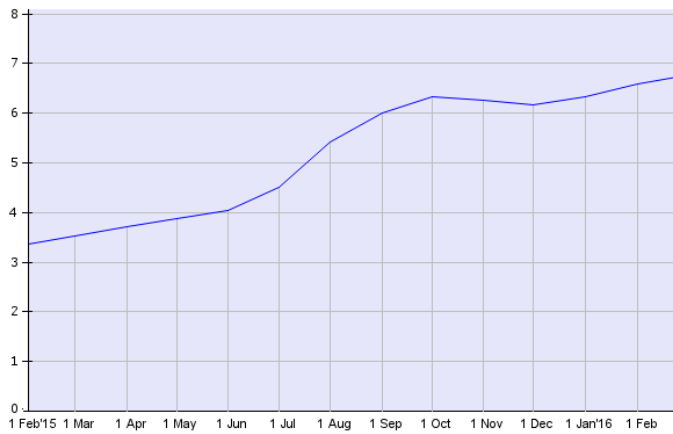
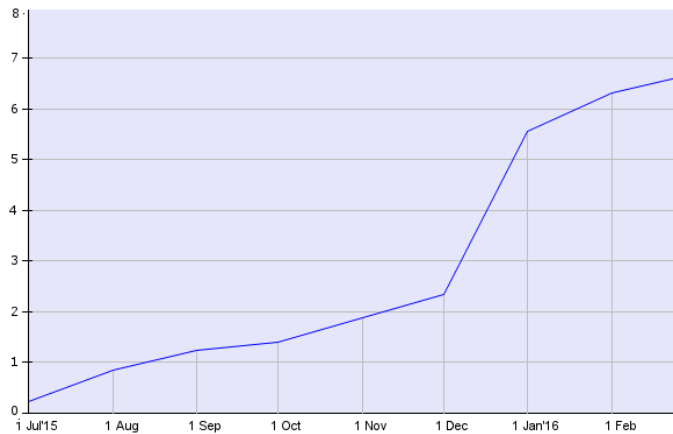Fig. 6. Usage of SPDY for Websites [13]

Fig. 7. Usage of HTTP/2 for Websites [14]

experimental results have suggested that the implementation of these protocols at the server end may perhaps improve the performance and reduce the web latency; however, the client support did not find any significant improvement in the performance. Indeed, web protocols are one of the major factors responsible for web latency, however, the wide variation (based on this observation a more thorough evaluation is needed) in load times for similar types of websites suggest that they depend on the contents of the website and the location of the server. These factors were not considered in this investigation. Finally, the experimental results and current acceptance level of SPDY and HTTP/2 protocols suggest that their employment in WoT will require substantial enhancement or perhaps replacement with equivalent lightweight and low latency protocols such as MQTT, CoAP and AMQP. In the future, it may be interesting to investigate the implementation of HTTP/2 protocol within IoT system.

REFERENCES

[1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[2] D. Pfisterer, K. Römer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kröller, M. Pagel, M. Hauswirth *et al.*, "SPITFIRE: toward a semantic web of things." *IEEE Communications Magazine*, vol. 49, no. 11, pp. 40–48, 2011.

[3] D. Guinard. (2016, January 23) Web of Things vs Internet of Things. [Online]. Available: http://webofthings.org/2016/01/23/wot-vs-iot-12/

[4] G. Baker and E. Arvidsson, "Let's make the web faster," 2010.

[5] I. Grigorik, "Making the web faster with HTTP 2.0," *Communications of the ACM*, vol. 56, no. 12, pp. 42–49, 2013.

[6] M. Belshe, R. Peon, and M. Thomson. (2015, May) Hypertext Transfer Protocol Version 2 (HTTP/2). [Online]. Available: https://tools.ietf.org/html/rfc7540

[7] Github.io. (2015) What is HTTP/2? [Online]. Available: https://http2.github.io/

[8] chromium.org. (2010) SPDY: An experimental protocol for a faster web. [Online]. Available: https://www.chromium.org/spdy/spdy-whitepaper

[9] B. Thomas, R. Jurdak, and I. Atkinson, "SPDYing up the web," *Communications of the ACM*, vol. 55, no. 12, pp. 64–73, 2012.

[10] J. Erman, V. Gopalakrishnan, R. Jana, and K. K. Ramakrishnan, "Towards a SPDY'ier mobile web?" *Networking, IEEE/ACM Transactions on*, vol. 23, no. 6, pp. 2010–2023, 2015.

[11] F. Toomey. (2015, April 15) Why latency management will decide the future of the IoT. [Online]. Available: http://www.wirelessweek.com/article/2015/04/why-latency-management-will-decide-future-iot

[12] J. Parkinson. (2015, April 8) Forecasting the future of the Internet of Things. [Online]. Available: http://ww2.cfo.com/forecasting/2015/04/forecasting-future-internet-of-things/

[13] W3techs.com. (2016, February 21) Usage of SPDY for websites. [Online]. Available: http://w3techs.com/technologies/details/ce-spdy/all/all

[14] ——. (2016, February 21) Usage of HTTP/2 for websites. [Online]. Available: http://w3techs.com/technologies/details/ce-http2/all/all