The Impact of SPDY on the Mobile Broadband Ecosystem and Value-Added Services (VAS)

June 2014

TABLE OF CONTENTS

1. 4G AMERICAS POSITION STATEMENT

The growing rate of encryption, especially SPDY-encrypted Hypertext Transfer Protocol (HTTP) over mobile wireless networks, has both immediate and long-term impacts on the mobile broadband ecosystem and wireless technology architectures. As the rate of encryption grows, Mobile Service Providers (MSPs) are forced to rethink their service offerings and value proposition, how they manage capacity and customer experience and how value-added services will be impacted.

The term SPDY, pronounced “speedy”, was established and trademarked by Google and is not an acronym. It was developed as an open networking protocol for transporting web content. SPDY encapsulates multiple HTTP flows in a TLS header, with particular goals of reducing web page load latency and improving web security. SPDY achieves reduced latency through HTTP header compression, stream multiplexing, and request prioritization. As of July 2012, the group developing SPDY has stated publicly that it is working toward standardization (internet draft). The first draft of HTTP 2.0 is using SPDY as the working base for its specification draft and editing. Implementations of SPDY exist in Chromium, Mozilla Firefox, Opera, Amazon Silk and Internet Explorer and will be included in the Safari release accompanying Apple's OS X Yosemite.

4G Americas’ position on SPDY is that of collaboration and coordination with the organization wholly supporting and endorsing the work of the newly formed Open Web Alliance (OWA) that is addressed in section 4.1 of this paper. 4G Americas believes collaboration of this important issue that affects the wireless ecosystem is needed in terms of: 1) Working with key stakeholders to identify the requirements for solutions that do not conflict with each other, 2) Coordination with privacy advocacy groups to promote secure communications and 3) Cooperation with regulatory bodies. The OWA is an important step in the process of bringing all significant important players to one alliance to cooperate on the issues surrounding SPDY. The broad support in both mobile and desktop browsers/clients, coupled with the rapid adoption by leading Internet destinations, changes the way MSPs think about middleboxes or Internet nodes that provide transport policy enforcement functions for both service offerings and network management. For those destinations that have not adopted SPDY support—other companies such as Google and Amazon are providing SPDY proxy and split browser architectures that extend SPDY benefits by proxying the entire Internet. At this point, it is safe to assume that encrypted transport of Internet traffic, including HTTP 2.0, will become the rule, rather than the exception. In fact, the Internet Engineering Task Force (IETF) recently chartered a working group named “Using Transport Layer Security (TLS) in Applications” to simplify implementations and increase the adoption of encryption for all application protocols.

For this reason, traditional MSP models of transparent middlebox insertion into Internet Protocol (IP) streams are becoming technically unviable and the business models related are becoming more difficult to support. MSP general network management solutions (load balancers, Network Address Translators, Hierarchical Quality of Service (QoS);, transparent Value-Add Services (VAS), such as content/URL filtering and video optimization; security services, such as Firewalls, Distributed Denial of Service (DDoS) and malware detection; and monetization opportunities, such as analytics and deep packet inspection (DPI), are impacted to varying degrees. In some instances, SPDY proxy functionality renders some of these middlebox functions entirely useless. Figure 1 illustrates the Gi Local Area Network (LAN) Services Domain which resides on the 3rd Generation Partnership Project (3GPP) Gi interface or Internet-facing side of the mobile packet core.

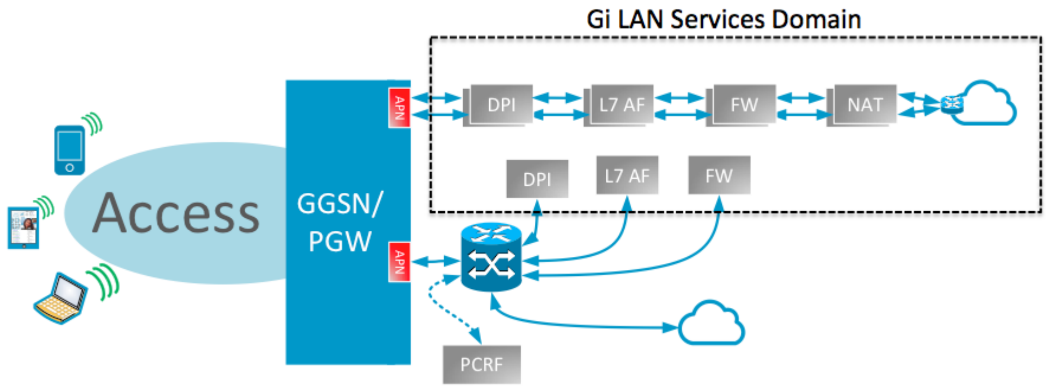

Figure 1. Gi LAN Service Domain.

The middlebox tasks include packet inspectors, content modifiers, protocol proxies, and Value-Added Services (VAS). Examples of such platforms and associated use-cases are identified in the following table.

Table 1. Middlebox: Service, Examples and Use-Cases.

Service	Examples	Use-Cases
Packet Inspectors[1]	• Deep Packet Inspection • Internet Protocol Flow Information Export (IPFix) • Network Probes • Network Firewalls • Web Application Firewalls • Network Security (Intrusion Prevention System (IPS)), Distributed Denial-of-Service (DDoS Detectors))	• Traffic/Activity monitoring • Data analytics • L2-L7 Load-balancing • Behavioral Analysis • Anomaly Detection • Denial-of-Service (DoS)/ Distributed Denial-of-Service (DDoS) Protection • Malware Detection
Content Modifiers	• Content Optimizers • Protocol Optimizers • Compression Engines	• Transrate/Compress Video • Image Compression • Optimize Transmission Control Protocol (TCP) Slow-start
Protocol Proxies[2]	• Network Address Translators • Domain Name System (DNS) Cache • Session Initiation Protocol (SIP) Proxy • Session Border Controller • HTTP Proxy • Web Real-time Communication (WebRTC) Gateway • TCP Proxy	• Modifying IP address information • Communications Control • Content Caching • Area Border Router (ABR) Index Modification • Performance Enhancing Proxy • L7 Application Functions
Value-Added Services	• Ad Insertion Engine • Header Insertion	• Advanced Advertising • In-stream header enrichment • Uniform Resource Locator (URL) Filtering • Parental Control

[1] Packet inspectors have multiple roles, including pure monitoring of traffic, gating/blocking traffic, or routing/forwarding traffic.

[2] There is a whole class of Gi LAN services called “terminating services”, which act as protocol proxies on behalf of some application. For instance, in a VoLTE environment, the SIP proxies and SBCs are discovered and explicitly addressed by the application.

These middleboxes function by inspecting or modifying various headers as traffic passes between the requester (client) and responder (server). The information that can be extracted depends on the middleboxes' ability to interpret the information in the headers and how deeply into the IP packet the middlebox is capable of looking. In the case of HTTP traffic for instance, the following information is available at each header layer:

- IP (Layer 3) Header: Source and Destination IP address. In some instances, destination IP address may be indicative of a particular domain; however, with the increasing consolidation of content through cloud hosting companies, blogs and Content Delivery Networks (CDNs), the IP address is not always descriptive of a particular destination host. With encrypted destinations however, most hosting companies generally seem to require a dedicated IP address

- TCP/UDP (Layer 4) Header: Source and Destination Port Number. Since applications listen on particular ports for traffic, port number is typically indicative of the protocol being transported at the application layer

- HTTP (Layer 7) Header: Application-specific information such as requested Host (domain name), URL, user-agent (browser type), cookie data from the HTTP Request, status code and Content Type from the HTTP Response

These middleboxes may be impacted by encryption as their ability to modify transport or application layer protocols, or to modify content itself, becomes more difficult. The effectiveness of these middleboxes directly impacts carrier ability to optimize and monetize their network traffic.

In many instances, the middleboxes operate transparently to the Internet destination, providing a service that Internet Service Providers (ISPs) offer their customers. One such example of this transparent operation is a Parental Control service in which specific Internet destinations are blocked for minors at the discretion of the subscriber.

Additionally, network monitoring for intrusion detection, Denial of Service (DoS) attacks, and other security risks are widely used to detect malware command and control traffic and network reconnaissance. Encryption can make this detection more difficult, if not impossible.

2. SPDY AND HTTP 2.0 ORIGINS

SPDY was initially developed by Google in mid-2009 with the primary goal to reduce the load latency of web pages by addressing some of the performance limitations of HTTP 1.1. Specifically to:

- Target a 50 percent reduction in page load time (PLT)

- Minimize deployment complexity

- Gather real performance data to validate the experimental protocol

To achieve the 50 percent PLT improvement, SPDY designed a more efficient use of the underlying TCP connection by introducing a new binary framing layer to enable request and response multiplexing, prioritization, and to minimize and eliminate unnecessary network latency.

By 2012, SPDY was supported by Chrome, Firefox and Opera, plus many large web platforms (e.g., Google, Twitter and Facebook) which were all offering SPDY to compatible clients. SPDY proved to offer great performance benefits and was on track to become a de facto standard through growing industry adoption. As a result, the Internet Engineering Task Force (IETF) HTTP Working Group (HTTP-WG) kicked off the new HTTP 2.0 effort in early 2012 to secure that the lessons learned from SPDY are applied to the official HTTP 2.0 standard.

2.1 ROADMAP: SPDY TO HTTP 2.0

SPDY was the starting point for HTTP 2.0; however, SPDY is not HTTP 2.0. An open call for HTTP 2.0 proposals was made back in 2012, and within the HTTP-WG, the SPDY specification was adopted as a starting point for future evolution of the standard. The drafted charter for HTTP 2.0 highlights the scope and the key design criteria of the protocol:

It is expected that HTTP 2.0 will:

- *Substantially and measurably improve end-user perceived latency in most cases, over HTTP 1.1 using TCP*
- *Address the "head of line blocking" problem in HTTP*
- *Not require multiple connections to a server to enable parallelism, thus improving its use of TCP, especially regarding congestion control*
- *Retain the semantics of HTTP 1.1, leveraging existing documentation, including (but not limited to) HTTP methods, status codes, URIs, and where appropriate, header fields*
- *Clearly define how HTTP 2.0 interacts with HTTP 1.x, especially in intermediaries*
- *Clearly identify any new extensibility points and policy for their appropriate use*
- *The resulting specification(s) are expected to meet these goals for common existing deployments of HTTP; in particular, web browsing (desktop and mobile), non-browsers ("HTTP APIs"), web serving (at a variety of scales) and intermediation (by proxies, corporate firewalls, "reverse" proxies and Content Delivery Networks). Likewise, current and future semantic extensions to HTTP/1.x (e.g., headers, methods, status codes, cache directives) should be supported in the new protocol.*

Thus, HTTP 2.0 will address the performance limitations of preceding standards, but it is also extending, not replacing, the previous 1.x standards. The semantics of HTTP are the same and no changes are being made to the offered functionality or core concepts such as HTTP methods, status codes, URIs and header fields; these changes are explicitly out of scope. The reason for the major revision increment to 2.0 is due to the change in how the data is exchanged between the client and server. To achieve the outlined performance goals, HTTP 2.0 adds a new binary framing layer, which is not backward compatible with previous HTTP 1.x servers and clients. Figure 2 illustrates HTTP 2.0 protocol.

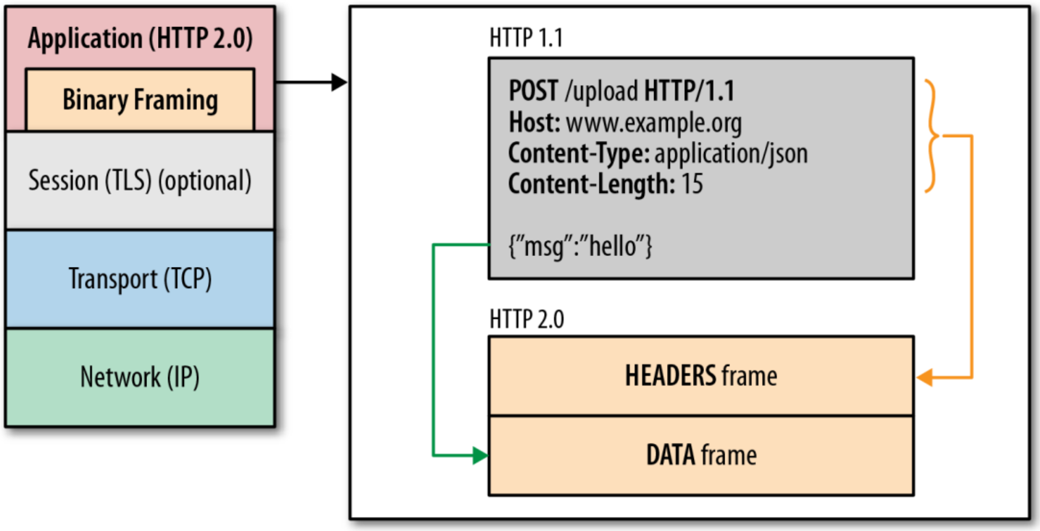

Figure 2. HTTP 2.0 Protocol.

The HTTP working group adopted the SPDY v2 draft as the starting point for the HTTP 2.0 standard, although both SPDY v3 (2012) and SPDY v4 (2014) are available. SPDY v4 aligns closely to the HTTP 2.0 standard. The timeline for HTTP 2.0, as per HTTP-WG official milestones, are to submit HTTP 2.0 to the Internet Engineering Steering Group (IESG) as a Proposed Standard by November 2014.

The switch to HTTP 2.0 cannot happen overnight; millions of servers must be updated to use the new binary framing and billions of clients must similarly update their browsers and networking libraries. However, most modern browsers use efficient background update mechanisms which will enable HTTP 2.0 support quickly. HTTP 1.x will be around for at least another decade and most servers and clients will have to support both 1.x and 2.0 standards.

The following diagrams provide web server context to the adoption of SPDY as a precursor to the HTTP 2.0 standards. As of May 2014, only 0.8 percent of web destinations supported SPDY; however, modules already exist supporting the top three web servers (Apache, NGINX, Microsoft-IIS). While the number of destinations is relatively low, Google, YouTube, Facebook, Twitter and WordPress all support SPDY already, representing five of the top 12 Internet destinations.

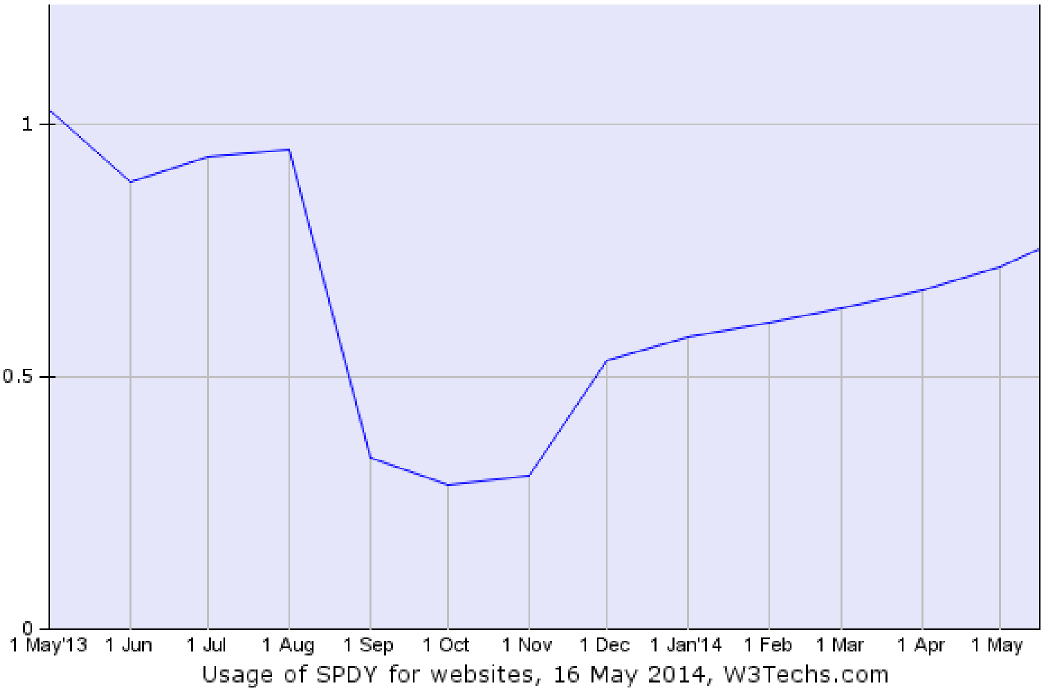

Figure 3. SPDY Website Adoption – Apr 2014.[3]

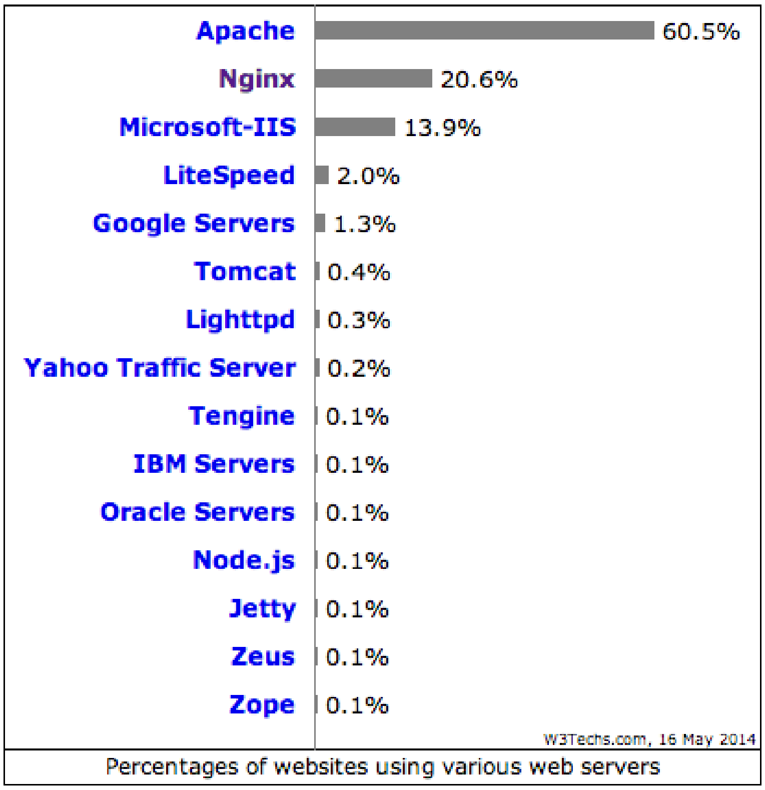

Figure 4. Web Server Adoption – Apr 2013.[4]

[3] W3Techs.com

[4] *Ibid.*

3. SPDY PROXIES AND THEIR IMPLICATIONS

While many of the largest Internet destinations have adopted SPDY, the inherent benefits over mobile networks (encryption, data compression and faster page load times), as well as the potential to capture an increasing set of analytics data or capture additional value, have prompted some organizations to implement SPDY Proxy functionality.

Like any other proxy, SPDY proxies act as an additional middlebox in the data path. The SPDY proxy terminates the SPDY session and carries on the HTTP session within the SPDY tunnel, extending support for SPDY for all destinations when requested by a supported browser. In addition, for websites that aggregate content from multiple destinations, a SPDY proxy multiplexes HTTP requests to multiple destinations over a single TCP connection, eliminating multiple TCP handshake overhead and TCP slow-start computations. Combined with the inherent benefits of SPDY, the proxy functionality reduces the amount of data consumed by the mobile subscriber.

A particular type of architecture where the functionality of the browser is offloaded to a cloud-based component in the form of a proxy is also known as split browser architecture. A detailed analysis of this architecture is presented in a white paper titled, *Exploiting Split Browsers for Efficiently Protecting User Data*.[5] Generally, split browser architecture requires the proxy to be configured and it has the full benefits when both sides are designed and developed by the same entity.

The predominant organizations providing SPDY proxy functionality are Google (for the Chrome browser included as part of the Android operating system) and Amazon (for the Silk browser used in tablets from that company). In both instances, the SPDY proxy functionality is hosted in the respective company's datacenters. The following figures illustrate the Amazon Silk split browser[6] and Google SPDY proxies[7].

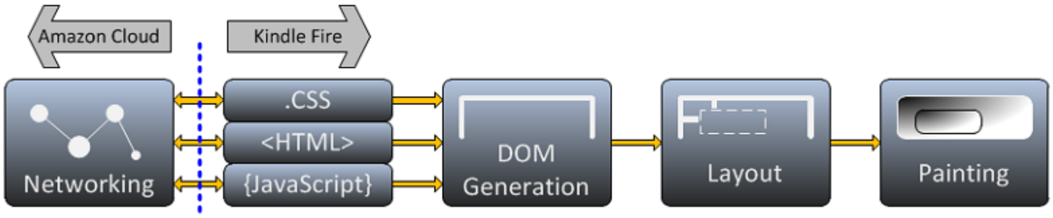

Figure 5. Amazon Split Browser SPDY Proxy.

[5] http://www.cs.columbia.edu/~angelos/Papers/2012/split_browser.pdf
[6] http://docs.aws.amazon.com/silk/latest/developerguide/split-arch.html
[7] https://developers.google.com/chrome/mobile/docs/data-compression

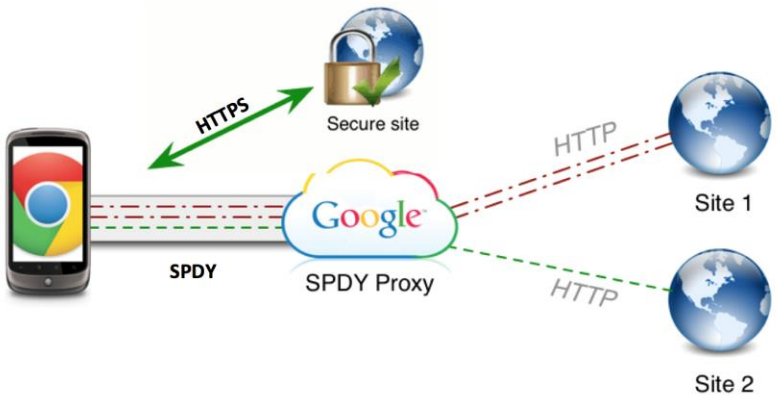

Figure 6. Google SPDY Proxy.

These proxies have negative implications for service providers from both a technical perspective and also have an overall effect on the entire mobile broadband ecosystem. Such negative implications include:

- Changes in dynamics in peering relationship negotiations with SPDY proxy owners, due to the increased inbound (to end users) and outbound traffic (from Communications Service Providers (CSPs))
- Reduced visibility into network traffic, including lost visibility into individual Internet flows and into DNS requests. In effect, the destination of all Internet traffic originated from a client that has a SPDY proxy configured in the SPDY proxy itself. This effectively eliminates the ability of a Service Provider to glean information of individual flows, whether encrypted or not
- Inefficient content routing, since the majority of CDNs rely on a combination of DNS Requests and Source IP Address to return IP ranges. In the case of the DNS late-binding mechanism used by SPDY proxies, the Source IP address issuing the DNS request would refer to the SPDY proxy (and the SPDY proxy's ISP), which might not be illustrative of the closest cache from which to serve content. This may directly impact the latency perceived by the client
- Traffic flow modification, potentially leading to different capacity planning models on a per-node and network-wide basis. SPDY changes flow sizes and durations, leading to fewer higher-bandwidth flows that impact network and infrastructure design
- Change the Value-Added Services (VAS) paradigm by: (a) removing the mobile broadband provider's understanding of access network conditions from decision-making; (b) coupling caching, video optimization, image compression and web acceleration with the SPDY proxy itself, and; (c) enabling a co-processing model for HTTP that eliminates well-known application protocols from traversing the service provider network

4. THE OPEN WEB ALLIANCE (OWA)

Several individual standards organizations, in addition to 4G Americas, have created initiatives to look at the implications of SPDY and HTTP 2.0 including their proxy functions and also to make recommendations on the path forward for both wireline and wireless Service Providers. The organization with the greatest momentum and impact is the Open Web Alliance[8] (OWA). The OWA charter reads as follows:

The Internet is a vibrant ecosystem based on reliable service delivery that supports innovation to provide a rich user experience—all through an intricate web of collaboration. Changes to the underlying protocols and architecture can have far-reaching technical and business implications. An Alliance for Telecommunications Industry Solutions (ATIS) initiative, the Open Web Alliance (OWA) recognizes that key decisions about Internet architecture are best handled through open, multi-stakeholder collaboration. OWA will enable richer, faster and more secure web services by addressing the challenges of proprietary proxies. The initiative will develop critical requirements for an open service optimization proxy that supports all stakeholder needs in the Internet service delivery value chain, including users' needs for encryption and privacy. Both the technical requirements and the business model implications for the realization of open service optimization proxies that benefit the whole Internet will be considered.

The three stated goals of the OWA in the realization of the above charter include:

- Collaboration with the broader ecosystem stakeholders to identify the requirements for solutions that do not conflict with each other and that do not infringe on the established trust relationships with the user. Today, solutions are being developed and implemented independently, without explicit consideration of the broader ecosystem impacts

- Collaboration with privacy advocacy groups to promote the use of secure communications and educate the user community on the various aspects of security including trust, user consent and transparency

- Collaboration with regulatory bodies in realignment with the current reality of Internet usage based on trends, technologies and service deployments

4.1 PATH FORWARD: 4G AMERICAS ENDORSEMENT OF THE OPEN WEB ALLIANCE

In order to ensure a unified view of future Internet architectures and reduce overlapping initiatives, it is the recommendation from 4G Americas to support the Open Web Alliance and its three goals stated above that were derived from the ATIS analysis on SPDY which can be found at: http://www.atis.org/openweballiance/docs/SPDY%20Analysis.pdf

ACKNOWLEDGEMENTS

The mission of 4G Americas is to advocate for and foster the advancement and full capabilities of the 3GPP family of mobile broadband technologies, including LTE-Advanced, throughout the ecosystem's networks, services, applications and wirelessly connected devices in the Americas. 4G Americas' Board of Governors members include Alcatel-Lucent, América Móvil, AT&T, Cable & Wireless, Cisco, CommScope, Entel, Ericsson, HP, Mavenir Systems, Nokia, Openwave Mobility, Qualcomm, Rogers, Sprint, T-Mobile USA and Telefónica.

4G Americas would like to recognize the significant project leadership and important contributions of Kevin Shatzkamer of Cisco, as well as representatives from the other member companies on 4G Americas' Board of Governors who participated in the development of this white paper.

[8] http://www.atis.org/openweballiance/index.asp