Recently, network performance measurement tools are available to end users to measure their network and diagnose problems. In particular, many measurement tools are built as a plugin on browsers. Although these browser-based network measurement tools have gained a great popularity, the accuracy of these measurement tools is little known by the users. Li et al. performed a delay accuracy measurement to these tools based on RTT(Round-trip Time) and reported their results.

To measure the latency, they use the term *delay overhead* (denoted as Δd) to indicate the difference between the value measured by the browser-based tools and the actual value (calculated through packet capturing using WinDump/tcpdump). t_s^B and t_r^B are the timestamp of sending a request and receiving the reply in browsers. t_s^N and t_s^N are the timestamp captured in the network stack of a client.

$$\Delta d = (t_r^B - t_s^B) - (t_r^N - t_s^N).$$

Two approaches -- HTTP-based applied with XHR, DOM, Flash, Java applet technologies and Socket-based applied with WebSocket, Java applet, Flash technologies are tested in five browsers. In term of methodology, they set up two machines with same hardware configuration with 100-Mbps Ethernet, one of the computers is the client and another one is the server(with crafted 50ms delay). Browsers in the client will send a request to the server and get reply message with the 50ms delay. Meanwhile, a Windump/tcpdump is running in the client to record the t_s^N and t_s^N . Considering the impact of browsers to instantiate objects, a second RTT management is conducted to get 2 delay overheads (denoted as Δd_1 , Δd_2).

As the result, all these approaches experience delay overhead in different degrees. In HTTP-based approach, XHR method's delay overhead ranges from a few milliseconds to tens of milliseconds. Overheads in Flash are the most significant, from 20ms to over 120 ms. DOM achieves better, most of the delay are less than 5ms. Java applet is different from other 3 technologies as it will cause under-estimate the RTT by as much as 5ms. On the other hand, delay overheads by Socket-bash approach methods are much smaller and more constant than HTTP-based approach. The median overheads are mostly less than 1 ms. WebSocket provides the most accurate and stable result. Also, java applet socket method introduces an underestimate delay.

In HTTP-based methods, unusual high delay overheads happened on Δd_2 when sending the second POST request. The reason for that is a new connection is still be built for the second POST request, while the connection can be reused for measurement in GET method. On the other hand, the underestimate incurred in Java applet is affected by the timestamping function *Date.getTime(),* experiments conducted by them show that the timestamp granularity caused by this function ranges from 1ms to 15ms and this granularity could last for a period of time. Delay overheads measured by Java applet methods in Windows when System.nanoTime() is used show that this underestimates disappeared. And the Java socket method has a same performance with tcpdump/Windump is System.nanoTime() is adapted.

All in all, delay overhead varies a lot depending on different methods and different browsers. Java applet socket method is recommended and Ubuntu Chrome is a better choice for browser-based network measurement tool. Future work can be done by exploring delay overhead on the server side or do the same experiments in smartphone environment.