Javascript Timing

Some of the measures were performed online using Javascript enabled web-pages running on participant's own equipment. Where local timing was required for such measures the Javascript Date.getTime() function was used and this, together with the uncontrolled nature of the remote equipment, introduces several caveats into the results.

1. Firstly, a participant's devices may update its internal timer while the test is running, for instance automatically synchronising with a remote time service. Intervals measured across such an update event will be incorrect. Time shifts resulting in an overall negative value for an interval can be identified when the data is received centrally, however shifts resulting in a small shift between two positive numbers cannot be reliably detected.

2. Within Javascript the precision of time measurement is coarser than the milisecond units used to return the results. As an example, under Windows-XP Javascript times are only returned to the nearest 1/64th epoch of a second, so two events in the same epoch appear simultaneous.

3. When supplying uniformly spaced stimuli to some browsers/devices the values returned by Date.getTime() are not uniformly spaced, with deviations of up to 0.3 seconds observed in testing. This effect was observed particularly often when testing with Internet Explorer, but may well be present in other environments.

In conclusion, timings returned via Javascript.getTime() cannot be regarded as reliable for sub-second processes but are generally acceptable for longer periods.