

## Understanding Internet Speed Test Results

### *Why Do Some Speed Test Results Not Match My Experience?*

You only have to read the popular press to know that the Internet is a very crowded place to work or play. Many Internet Service Providers (ISPs) are talking about changing from a monthly subscription model to a **usage** model, much in the way that analogue public telephone service has changed over time. It is not surprising, therefore, that most Internet users encounter performance related service problems on a fairly regular basis and want to validate that their ISPs are delivering the contracted service performance.

*The problem is not the in the measurement, it is in understanding the test results as they relate to the application problem being experienced.*

As a result there is a multitude of speed testing websites to be found on the Internet. Some of these testing services are provided by the ISP, while many are provided by independent third parties. Naturally the key question is, "Is the speed test accurate?" And if the speed test results do not meet expectations, does the tester provide the data to resolve the unexpected results?

Unfortunately, all speed tests are not created equal and the testing applications rarely define the method used to conduct the test. Without understanding the testing methodology, validating the results is a difficult or impossible task for the user, regardless

of their skill level. The complaint most echoed around the Internet discussion groups is, "Why is this video download taking so long when my speed tester shows I get my full 10Mbps?"

The problem is not the in the measurement, it is in understanding the test results as they relate to the application problem being experienced.

A better understanding of Internet speed test methods can be gained through comparisons to vehicular traffic. For example, if the local airport is 60 miles away and the road speed to the airport is 60 miles per hour, if you put 4 passengers in your car and drove to the airport you would most probably express that speed as 4 passengers per hour. If you now rented a bus and filled it with 50 passengers and drove the exact same journey, you could report your speed as 50 passengers per hour. However, the local authority that owns the road to the airport might publish the road's capability as 50,000 passengers per hour. This difference lies in local authority's assessment of passenger throughput for the road at maximum capacity -- with all the passengers in all the cars and buses added together, then using that total to express the passengers per hour of the road. Each of these 3 different measures has validity within the context of the test conducted. However, each test method is completely different in the

value it delivers for the individual application user. One of the biggest failures of Internet speed testing applications is their inability to differentiate a true **application** speed test from a less meaningful **capacity** speed test. What is the difference?

As an example, a local telecom provider announced in the major press recently that they would soon offer 100Mbps (100 megabits per second) connections to the home and download speeds will be 20-30 times what they are today. Examples were quoted of a movie download taking only minutes whereas existing services are measured in hours. The problem with this statement is that it can be completely misleading -- just because an Internet connection is rated as 100Mbps does not mean you will get a 100Mbps speed on your connection.

Just like the simple road example above, where the local road authority published the speed at 50,000 passengers per hour, the speed of a connection is being published at 100 megabits per second. The problem with speed testers that measure capacity speed, as some of the most popular testers do, is that the **capacity** of a connection does not relate to the **application speed** of the connection. The result of this disconnect is that the user of the connection gets very frustrated with the actual speed achieved because reality does not match the published expectation.

In the road speed test example, the bus application was only 50 passengers per hour, the car's was only 4 passengers per hour, the reason for this is obvious. However, to understand the importance of an **application speed test** it is first necessary to understand some of the principles of why the Internet is designed the way that it is.

First, the Internet is often described as a **contended** network as well as a **best effort** network. A **contended network** means that all the users contend for use of the Internet highway, not unlike cars contending for use of the traffic highway. **Best effort** means that in a contended network there are no guarantees that your application data will get to the destination in a timely manner, or even at all.

With these limitations in mind, the Internet was designed to cope with the stresses and strains of contention. However, to do this meant that the protocols that drive the internet had to incorporate **flow control**. Without some element of flow control the Internet would simply not work, it would collapse under the stresses of the data as fast connections joined slower connections. In our road speed example, if you are driving to the airport with your 4 passengers and arrive at a junction that has very heavy traffic, your ability to enter that traffic flow will be dependent on there being a gap between two vehicles, or possibly a set of traffic lights that give you priority at a certain point in time (regulation). Of course if traffic is so heavy at the junction that there are no gaps, and there are no regulatory lights, then it is unlikely that you will achieve 4 passengers per hour and you may not get to the airport at all.

In the Internet world bytes are not measured in values as low as 4 or even 50, as in the car/bus example. Instead the Internet deals with numbers of data bytes that are many orders of magnitude greater. For example, a download of a music file can be measured in hundreds of millions of bytes. To help resolve the two main issues of **contention** and **best effort** the Internet management process sends data in limited chunks at any one time, after which the sending computer waits to hear from the receiving computer that the data has arrived before sending more data. This procedure allows the process to ensure the integrity of the data, as well as recovery if data is lost. Taking this approach delivers two very important characteristics to performance: 1) The performance of the connection must include the return journey time back to the starting point because confirmation is required before more data can be sent, and 2) The segmentation of the data chunks in this manner eases traffic congestion by effectively creating natural **gaps** in the traffic.

Given this consideration and applying it to the airport car example, if you had more than 4 passengers to drive to the airport in your car, the *passengers per hour speed* would not be 4 as previously stated, but 2 per hour as the total journey time would need to include the hour to the airport, and an additional 1 hour return journey to collect the next lot of passengers. A key point to understand in this analogy is that the performance throughput of any connection will largely depend on the distance between the starting point and the destination, coupled with the size of the vehicle used to carry the passengers. The bus, for example, would provide better passengers per hour performance because it could take 50 passengers at a time.

*The performance throughput of any connection will largely depend on the distance between the starting point and the destination, coupled with the size of the vehicle used to carry the passengers.*

However we need to delve further into Internet performance characteristics to better understand measurements of throughput and speed.

At this point you might think *great*, then we will have a bus that can take millions of passengers at a time, not a car. This is a possibility, in theory, however this approach will be impacted by contention because the chunk of data would be so much larger. In addition there is the issue of what happens when the vehicle does not make it to the airport because of the resulting contention issues. In the Internet

world this is called **packet loss**. Each chunk of data sent is broken into smaller chunks called packets and packets may not reach the destination because of contention issues.

*When packet loss occurs, chaos reigns.*

When packet loss occurs, chaos reigns. The receiving end has to notify the sending end that a packet has not arrived. There can be many different reasons for this, but regardless of the reason the sending end has to be told to resend the missing packet(s) of data. The amount of chaos caused will depend on where the lost packet was in the chunk of data and just how many packets were lost.

One very important reason for the chaos with regard to the application is that **data has to be processed in order**. If a packet is lost at the beginning, then the receiving end cannot process the subsequent packets until the missing packet or packets are recovered. So those packets that follow the missing packet must be stored until the missing packets are resent and the application that wants the data has to wait. This can happen several times with the same chunk of data, so the larger the chunk the larger the risk.

You might now think, rather than send one very big bus of passengers let's send two smaller buses or even more, let's send 4 busses. Unfortunately, this does not address the problem because most critical Internet applications require the data to be received in order and bus number 2 could arrive before bus number 1. Real-time video is a good example, imagine watching a video when frame number 100 appears before frame number 10. For video applications and even financial applications, such as stock trading, data must be processed in the order sent for the application to function correctly. Some applications can support multiple buses to move data but applications that can accept data in any order and still function correctly are not that common. Web pages are a good example of an application that can accept data out of order, for this reason it is not uncommon for items at the bottom of the page to appear before items at the top. However this only works because it does not affect the web page usage.

A vital question then is, "Why does this matter to a speed test?"

It matters a *huge* amount, and therein lies the problem of understanding the results of a speed test.

The Internet delivers a wide range of applications to the user, be it listening to music, watching a video, browsing a website or trading stocks. Each application will make use of the connection differently and the performance achieved will be subject to the application requirements and usage model. A speed test that does not invoke a test method that matches the application usage will not deliver a measure that will reflect the actual performance of the connection as it relates to that application. This oversight by speed testing applications is singularly the cause of more user frustration than any when trying to understand:

"Why is this video taking so long to download when my speed tester shows I get my full 10Mbps?"

Let's now examine the different types of speed test that exist and why they mostly fail.

## The Capacity Test

One of the most popular test applications on the Internet invokes a **capacity test**. In other words it is guilty of sending multiple buses to the airport all full of passengers. As a result the testing application effectively fills every lane on the user's Internet highway. The testing application can take this approach

*A speed test that does not invoke a test method that matches the application usage will not deliver a measure that will reflect the actual performance of the connection as it relates to that application.*

primarily because it does not care about the distance travelled, the packet order or even the packets that get lost because of the increased contention. The test simply reports the capacity speed of the connection based on the volume of data that survives the journey and arrives at the destination. As the speed testing application does not care about the order of the data or the trip time for a connection, it delivers an incorrect and misleading result. This is the equivalent of the airport road authority publishing a speed of 50,000 passengers per hour.

Unfortunately, unless the testing user realizes that the test application is sending multiple buses of data simultaneously to provide a capacity speed measure, they will get very frustrated when their video application, which has to maintain data order and recover all lost packets to provide a quality movie experience, fails to match the performance level reported. Hence the question:

"Why is this video taking so long to download when my speed tester shows I get my full 10Mbps?".

## The Compression Test

Another popular method used in speed testing applications is to provide an HTTP download of a large file, usually a jpeg image or equivalent. This is a simple but common approach. The testing logic reads and times the data and reports the speed accordingly. On the surface you would expect that expressing data over time would deliver good results, however, because the jpeg file can be compressed the amount of data sent may be smaller than the data being read. Compression ratios can be as high as 40% or more and therefore deliver a faster reading than is actually achieved by the same percentage. The result of this variance is not visible to the testing user except when an application such as a video download starts taking longer to download because a video cannot be compressed to the same extent if at all.

## The Timing Error Test

The timing of Internet connection speed requires an accurate timer, this generally goes without saying. Unfortunately in a multi-tasking, time-sharing PC environment this is easier said than done. And faster connection speeds demand higher accuracy of measurement. As an example of bad timing, one speed testing application recently provided a report listing the data transferred per second with a summary of the overall rate per second at the end of the report. However, **timing on a per second basis is not accurate**. To clarify this point further, many years ago it was common to time an athletic running event using a pocket stopwatch which timed in seconds. Now imagine timing the 2008 Olympics 400 meter freestyle when Michael Phelps won by just 1000th of a second using the same stopwatch. The result would have been entirely different, most probably a draw. When Internet speed is measured in millions of bits per second, timing the data to at least 1000th of a second is essential to maintain accuracy of the result. The challenge is maintaining an accurate timer to 1000th of a second on a PC platform. Failure to do this (and validate that this is done) can result in a 95Mbps test reporting 85Mbps.

## The Route Variance Test

One of the most common problems heard from ISPs, who deal daily with a multitude of 'speed' related issues, is that their customers use any random speed tester they find on the Internet and accept the results without any regard to location relevance or distance.

It is important to note that distance is not measured in miles but in time. Just as two roads that lead to the same city will be unlikely to provide the same travel time, no two Internet routes to the same geographical location will deliver the same route time. The only distance measure that matters is the **trip time**.

We have already discussed that distance to a destination is critical in understanding and accepting application speed results. That being the case, when a speed test is run that provides no feedback on trip time, or if that distance measure is ignored, then the speed result is meaningless. In the original airport car analogy, if the airport was not 1 hour away but 1 minute away, then the 2 passengers per hour speed rating would have increased to 120 passengers per hour (i.e. 4 passengers every 2 minutes, not 4 passengers every 2 hours).

Quite often speed test users will test their connection using more than one speed testing application that are in entirely different worldwide locations, and then question why the results are different.

## Conclusions Summary

The conclusions to derive from this discussion on speed testing are that:

1. Accuracy is only as good as the testing application's ability to provide the details that validate the test results.
2. Not all speed testers are equal and should not be compared unless the differences are taken into consideration.
3. The results from a speed tester should declare the method used to conduct the test, otherwise the results are meaningless.
4. The test method must align to the application profile for the results to be of value when assessing the performance expectation of that application.