Internet Society

The ISP Column
An occasional column on things Internet

It’s the Latency

May 2004

One of the more remarkable developments in physics occurred at the start of the twentieth century when the classical world of physics confronted some rather strange experimental outcomes. These experiments, dating back to the 1880's, coupled with Maxwell's electromagnetic theories of light, slowly convinced the scientific community that irrespective of the relative motions of a light source and an observer, the speed of light remained a constant value. Einstein embraced this view in his special theory of relativity, and postulated that the speed of light is indeed a universal constant, and, surprisingly concluded that space and time are not constants within our universe. This lead to the downfall of Newtonian physics as a base for universal physics, and heralded a new impetus in physical sciences as we explored the far reaching implications of this theory.

The speed of light also has some profound implications for networking technology as well. Light, or electromagnetic radiation, travels at 299,792,458 meters per second in a vacuum. Within a copper conductor the propagation speed is some three quarters of this speed, and in a fibre optic cable the speed of propagation is slightly slower, at two thirds of this speed.

Why is this relevant to network performance?

While it may take only some nanoseconds to pass a packet across a couple of hundred meters within a local area network, it takes considerably longer to pass the same packet across a continent, or across the world. In many parts of the world Internet services are based on satellite access. Often these satellite-based services compete in the market with services based on terrestrial or undersea cable. And even with undersea cables, the precise path length used can also vary considerably.

What's the difference?

The essential difference is "latency", or the time taken for a packet to traverse the network path. Latency is a major factor in IP performance, and reducing or mitigating the effects of latency is one of the major aspects of network performance engineering. And the time taken to get a packet through a network path is dependant on the length of the path and the speed of propagation of electromagnetic radiation through that medium. And this delay becomes a very important factor in network performance.

Why is Network Latency so Important for TCP/IP

In order to understand why this is the case it is appropriate to start with the IP protocol itself. IP is a 'datagram' protocol. For data to be transferred across an IP network the data is first segmented into packets. Prepended to each packet is an IP protocol header. IP itself makes remarkably few assumptions about the characteristics of the underlying transmission system. IP packets can be discarded, reordered or fragmented into a number of smaller IP packets and remain within the scope of the protocol. The IP protocol does not assume any particular network service quality, bandwidth, reliability, delay, variation in delay or even reordering of packets as they pass through the network.

In itself this unreliable services sounds like its not overly useful. If sending a packet through an IP network is an exercise in probability theory rather than a reliable transaction, then how does the Internet work at all?

If we want reliable service then we need to look at a protocol that sits directly on top of IP in a stack-based model of the network's architecture. The real work in turning IP into a reliable networking platform is undertaken through the Transmission Control Protocol, or TCP. TCP works by adding an additional header to the IP packet, wedged between the outer IP packet header and the inner data payload. This header includes sequencing identifiers, fragmentation control fields and a number of control flags.

TCP is a reliable data transfer protocol. The requirements for reliability in the data transfer implies that TCP will detect any form of data corruption, loss, or reordering on the part of the network and will ensure that the sender retransmits as many times as necessary until the data is transferred successfully. The essential characteristic of TCP is that the sender keeps a copy of every sent packet, and will not discard it until it receives an acknowledgement (ACK) from the other end of the connection that the packet has been received successfully.

Of course sending data packet by packet and waiting for an acknowledgement each time would be a very cumbersome affair. TCP uses 'sliding window control' to send data. By this, it is meant that the sender sends a sequence of packets (a 'window'), and then holds a copy of these packets while awaiting ACKs from the receiver to signal back that the packets have arrived successfully. Each time an ACK for new data is received, the window is advanced by one packet, allowing the sender to send the next packet into the network. The packets are numbered in sequence so that the received can resort the data into the correct order as required.

TCP also uses a form of adaptive rate control, where the protocol attempts to sense the upper limit of sustainable network traffic, and drive the session at a rate that is comparable to this maximal sustainable rate. It does this though dynamically changing the size of the sliding window, coupled with monitoring of packet loss. When TCP encounters packet loss, as signalled by the ACK return packets, it assumes that the loss is due to network congestion, and the protocol immediately reduces its data transfer rate and once more attempts to probe into higher transfer rates. This control is undertaken in TCP by adjusting the size of the sliding window. The way in which TCP adjusts its rate is by increasing the window each time an ACK for new data is received, and reducing the window size when the sender believes that a packet has been discarded.

There are two modes of window control. When starting a TCP session a control method called "slow start" is used, where the window is increased by one packet each time an ACK is received. If you get an ACK for every sent packet, then the result is that the sender doubles its window size, and hence doubles its data transfer rate, for each time interval required to send a packet and receive the matching ACK. Networking is not a science of the infinite, and a sender cannot continuously increase its sending rate without limit. At some stage the receiver will signal that its receiving buffer is saturated, or the sender will exhaust its sending buffer, or a network queue resource will become saturated. In the last case this network queue saturation will result in packet loss. When TCP experiences packet loss the TCP sender will immediately halve its sending rate and then enter "congestion avoidance" mode. In congestion avoidance mode the TCP sender will increase its sending rate by one packet every round trip time interval. Paradoxically, this rate increase is typically very much slower than the "slow start" rate.

For TCP, the critical network characteristic is the latency. The longer the latency, the more insensitive TCP becomes in its efforts to adapt to the network state. As the latency increases, TCP's rate increase becomes slower, and the traffic pattern becomes more bursty in nature. These two factors combine to reduce the efficiency of the protocol and hence the efficiency of the network to carry data. This leads to the observation that, from a performance perspective and from a network efficiency perspective, it is always a desirable objective to reduce network latency.

Of course these always a gap between theory and practice. The practical consideration is that TCP will only run as fast as the minimum of the size of the sending and receiving buffers in the end systems. On longer delay paths in the Internet its not the available bandwidth in the network thats the bottleneck, its the performance of the end system coupled with the network latency that limits performance.

What happens when I want to get a large file?

Lets look at this in the context of large data transfers.

Increasingly, the Internet is being used to pass large data files. These may be formatted office documents, mp3 multimedia files or large data sets. Such data objects form the bulk of the volume of traffic passed across IP networks, often accounting for more than 80% of the current volumes of data moved across the network. These data objects range in size from 1 - 10 megabytes (formatted documents) through to 100 megabytes (MP3 video files) and gigabytes (large data sets).

While end systems can be tuned to support high speed data transfers, most systems are not tuned for high speed long delay network paths, and the default settings in most platforms use a default local buffer size for TCP that ranges between 16KBytes and 64Kbytes. The implication of the local buffer limitation is that once the sending system has sent a full buffer of data, it can send no more data until the first ACK arrives back. The maximal data rate that can be sustained by a system is therefore one buffer size per round trip time.

Assuming a lossless path (no packet drops), and looking at a range of latencies, it is possible to provide a table of the "best possible" performance of a file transfer for a range of file sizes. The table below indicates this calculation for a number of common Internet latency times. Path A is a land path across a city, path B is a land path across a continent, while path C is an undersea path across a large ocean, such as the Pacific. Path D is a traversal across hemispheres and across an ocean, and path E is a path using a geostationary satellite. The buffer size of 16Kbytes is a common one used by default in many end user systems today.

		PATH				
		A	B	C	D	E
latency (ms)		1	15	75	130	334
RTT (ms)		2	30	150	260	668
Buffer Size	**16KB**					
Transfer Times						
File Size:	**5MB**	1s	9s	46s	1m 21s	3m 29s
	100MB	12s	3m 7s	15m	27m	1h 9m
	1GB	2m	31m	2h 36m	4h 31m	11h 35m

Table 1. "Best" Performance of various network paths for data files transfers using a 16KB TCP Buffer Size

The results are surprising - what takes as little as 2 minutes across town takes more than 4 hours on a high speed long distance submarine cable system, and 11 hours by a satellite path. And this assumes that there are no network bandwidth issues, and there is a perfectly lossless data path for the entire duration of the transfer.

In comparing networks paths for their effects on performance of user applications in long delay scenarios, the differential factor is not related to the variance in the round trip times, but their relative ratio. Doubling the round trip time for longer delay paths results in halving the potential transfer rate.

If you cannot reduce latency, and you are at the other end of one of these longer data paths what can you do? The basic answer is: equip your system with as much memory as you can scrounge, and tune your system to use very large buffers. Tuning a TCP stack to support window scaling, selective acknowledgements and large TCP buffers is a very effective means of limiting the worst effects of latency. And you also have to hope that the system at the other end has made similar adjustments to their TCP configuration. You cannot make light go faster, nor can you shrink the globe, so latency, and its effects are a constant factor here, but you can tune the servers and clients to reduce the effects of latency.

Can I speed up my Web Browser?

Not all Internet transactions move large data files around. The picture for short transfers is somewhat different.

While the majority of the traffic on public Internet networks today is due to large TCP data transfers, the bulk of individual transactions remains web-based page pulls. Web pages typically are between 10KBytes to 200Kbytes in size, and this smaller file size brings into consideration another aspect of the TCP transport protocol, plus application behavior.

The Web application first has to translate a URL into a target IP address. This is achieved through the use of DNS resolution. The DNS transactions are variable in time, predominately due to the level of local caching of previous queries. A reasonable assumption is that the DNS will require 2 round trip times to execute queries to resolve a URL to an IP address.

Once the IP address is established a TCP session is started. The three- way initial TCP handshake takes 1 and a half round trip times to complete. The requesting system then passes the data request, and the server processes this request and commences data delivery. At this point TCP operates in "slow start" mode. The first round trip sends one packet (approximately 540 bytes). The acknowledgement then allows the sender to send a further two packets (1080 bytes) back-to- back. The general sequence here is that TCP doubles its sending rate each round trip time interval, and will continue to do so until the systems' buffer space is exhausted, or until packet drop is encountered. Using a 16KByte TCP buffer, this algorithm will take 17 round trip times to complete the data transfer and a further round trip interval to handshake on TCP session closure, assuming optimal performance. The total number of round trip time intervals is 21 to complete the transaction.

This equates the 'responsiveness' of the network, in that the elapsed time from clicking a URL to having the page drawn on the screen will take approximately 21 round trip time intervals to complete.

Table 2 shows this elapsed time for same set of network paths used in the previous table.

		PATH				
		A	B	C	D	E
latency (ms)		1	15	75	130	334
RTT (ms)		2	30	150	260	668
Buffer Size	**16KB**					
Transfer Times						
File Size:	**200K**	0.04s	0.6s	3s	5s	14s

Table 1. "Best" Performance of various network paths for 'Web Responsiveness'

This is the 'responsiveness of the network - the minimum time taken from the time of the 'click' on a web page until the page is fully drawn What happens in the blink of an eye across town takes a slow breath across the world via cable. Again its the relative ratios of the paths that allow you to compare two paths for response to such traffic, not the absolute difference of their latency measures.

Tuning your system can help a little to improve things. For a server, one option is to increase the initial window size for TCP sessions to 4 segments rather than a default value of 1. It may not be much, but you save on 2 round trip intervals for short TCP transactions. Also lifting the TCP buffer size to a larger value may also be of some further benefit, shaving a further 4 round trip intervals from the time for a transaction of the type in Table 2, assuming the other end has also made similar modifications.

What about a new Transport Protocol?

There used to be a common cry in the area of network engineering, that every problem we faced was another facet of just not having enough bandwidth in the network. Advances in fibre optics over the past decade have all but banished this problem from the core of modern IP networks. And now it seems that in terms of TCP/IP, the speed of light, as expressed through latency within the network, has a significant impact on network performance.

One of the challenges we face today is that once you move of of the context of a local network and want to move massive data sets from one part of the globe, where, perhaps the observatory is located, to another, where the supercomputing services are housed, TCP as we know it, is nowhere near fast enough to fill up a 10Gbps longhaul IP network. And this time its not the bandwidth, not the bit level error probability, not any jitter factors, but the speed of light through fibre cable, and the geometry of the earth that TCP is struggling against.

We will not give up on this problem easily, and already the research community has made a number of very intriguing proposals for gigabit per second transport protocols over high delay paths, some of which we can explore in a future column. Its likely that we'll see such very high speed transport protocols evolve in the coming years, and I'm confident that we'll see deployment of such approaches to making long delay IP paths deliver far better network performance, as long as you are playing with moving large data volumes.

Of course there's always Plan B in any case. Our Plan B options are to either increase the speed of light, or be very patient and wait for continental drift to pull us all closer together. And the general consensus today is that the first option is not possible in our particular universe!

Geoff Huston

Disclaimer

The above views do not represent the views of the Internet Society, nor do they represent the views of the author's employer, the Telstra Corporation. They were possibly the opinions of the author at the time of writing this article, but things always change, including the author's opinions!

About the Author

GEOFF HUSTON holds a B.Sc. and a M.Sc. from the Australian National University. He has been closely involved with the development of the Internet for the past decade, particularly within Australia, where he was responsible for the initial build of the Internet within the Australian academic and research sector. Huston is currently the Chief Scientist in the Internet area for Telstra. He is also the Executive Director of the Internet Architecture Board, and is a member of the APNIC Executive Committee. He was an inaugural Trustee of the Internet Society, and served as Secretary of the Board of Trustees from 1993 until 2001, with a term of service as chair of the Board of Trustees in 1999 and 2000. He is author of a number of Internet-related books.