The Ultimate Online Experience™

Transmission Control Protocol (TCP): *It's Time for Something Better*

TCP has become a limiting as opposed to enabling factor when it comes to supporting today's web-facing applications. Learn how Internap's new Accelerated IP service works to bypass the protocol's shortcomings and can improve the performance of your applications and the experience of end-users.

An Internap White Paper

Table of Contents

TCP Defined

The bulk of Internet traffic today is facilitated by the Transmission Control Protocol (TCP). TCP sits between the Internet Layer and the Application Layer of the Internet Protocol Suite and supports the transmission of data between web servers and web browsers. Common web-based, Application Layer protocols such as Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS) and File Transfer Protocol (FTP) rely on TCP to accomplish their own communication objectives.

TCP was born out of a defining paper published in 1974 for the Institute of Electronic and Electrical Engineers that outlined a recommended structure for processing packet delivery over a data network. Since bandwidth and network congestion were not the concerns of the day, but reliability of the connection was, the defining recommendations were heavily focused on ensuring confirmed delivery of the entire set of packets.

Shortcomings of TCP

TCP was first put into practice in the early 1980s when the overall standard for the combined TCP/IP structure was set. Over time, with its focus on reliability and successful delivery at the expense of efficiency, TCP has become a limiting factor in the quest to utilize and extract more value from the increased bandwidth and enhanced processing capabilities that define today's physical Internet.

The protocol's underlying approach to supporting communications between an application and a client is simply not suited to the complexities or the capabilities of the networks supporting businesses today. TCP's approach to waiting, verifying the delivery of packets, slowing and then stopping and restarting does not match and/or support the needs and expectations of both application owners and end users.

TCP's dated design will only continue to prove problematic with each passing innovation, new way of connecting and increased expectation regarding the interaction with a web-based application. TCP has become the equivalent of the elementary school hall monitor – always counting and re-counting heads while barking *"single file, stay together and DON'T RUN!"* The problem is…the underlying infrastructure that is now the Internet has grown up, matured and needs to be free of these simple restrictions to operate more efficiently.

Analysis of TCP

TCP processes packet delivery according to a simple, three phase approach: establish a connection, transfer the data and then terminate the connection. The bulk of the intelligence and current issues with the protocol lie within the architecture of Phase II: data transfer. TCP's performance in this phase is limited by its obsession with not overloading either the receiver or the network.

	Phase I	Phase II		Phase III
Objective	Establish a Connection	Transfer the Packets		Terminate the Connection
Approach		Establish the Rate of Flow via *Flow Control*	React to Network Congestion via *Congestion Control*	
Tool(s)	3-Way Handshake	RWND	Slow Start Congestion Avoidance Fast Retransmit Fast Recovery	FIN Segments
Changes			Congestion Avoidance Algorithms a.k.a. TCP Variants TCP Reno CUBIC Compound TCP	

Figure 1: A Descriptive View of TCP

TCP initiates the flow control process by identifying the amount of data the receiving application can handle. This measurement – identified as the receiving window or RWND – states the maximum amount of data that can be received at one time. The sending application is limited to this amount until it receives an update from the client. The purpose of this exchange is to match the transmission rate of the sender to the receiver and ultimately to the network. Once this setting is established, TCP proceeds with the transfer of data packets but throttles its transmission rate according to an ongoing assessment of the congestion level of the network via the use of Congestion Control. Congestion Control takes advantage of four protocol subsets that help to govern the rate of packet transmission: Slow Start, Congestion Avoidance[1], Fast Retransmit and Fast Recovery.

---

[1] Congestion Avoidance is a core part of the current version of TCP because of an episode in the mid 1980s where the Internet experienced its first "congestion collapse." At that time, TCP did not account for congestion on the network and therefore it did not govern the rate of transmission. As a result, the first TCP variant (TCP Tahoe) was created and the Congestion Avoidance Algorithm was introduced.

Slow Start dictates the rate at which new packets may be sent to the receiver and is managed by the sender-delivered Congestion Window (CWND). CWND is essentially a means by which the receiving or client side can communicate its buffer capacity to the sending application. Each time a successful Acknowledgement (ACK) is received, Slow Start will allow the sending application to increase its transmission rate by one segment.

TCP continues to ramp up its allowed rate of transmission – attempting to reach a ceiling dictated by the combined RWND, CWND and the rate and success of receiving ACKs – while abiding by the parameters of the Congestion Avoidance Algorithm. Congestion Avoidance is guided by the number, manner and timing in which it receives ACKs. If duplicate ACKs are received, indicating that packets are arriving out of order, the Congestion Avoidance Algorithm will call for TCP to half its transmission rate and proceed at that level until the resulting ACKs indicate that sequencing is back on track. If the Congestion Avoidance Algorithm measures a timeout before receiving an ACK, implying that perhaps a packet has been lost, it will call for TCP to invoke Slow Start and thus dramatically reduces the rate of transmission. This governing overlay within TCP consistently creates a transmission pattern or rate that mimics the teeth on a saw – up and down, and up and down again.

TCP Reno – the most prevalent TCP Variant in place today – itself is now 20 years old and is woefully inadequate in matching the capabilities of networks and applications. The variant is too reliant on packet loss (i.e. getting duplicate or no ACKs) to assess congestion and thus prevents applications (and end users) from taking advantage of the bandwidth available and the advanced processing capabilities of attached network hosts. The reaction to halve or completely reset the transmission speed is an overreaction to a lost packet. Retransmission is warranted in this instance, but not a complete restart or, at best, half of the previously-achieved throughput rate. For practically every TCP-facilitated transmission, the process is similar to trying to win a race by slowly pressing on the accelerator at the start, riding the brake throughout the race, indiscriminately slamming on the brake and then gradually accelerating again. The outcome of this approach is unlikely to ever be a waving checkered flag.

The Solution

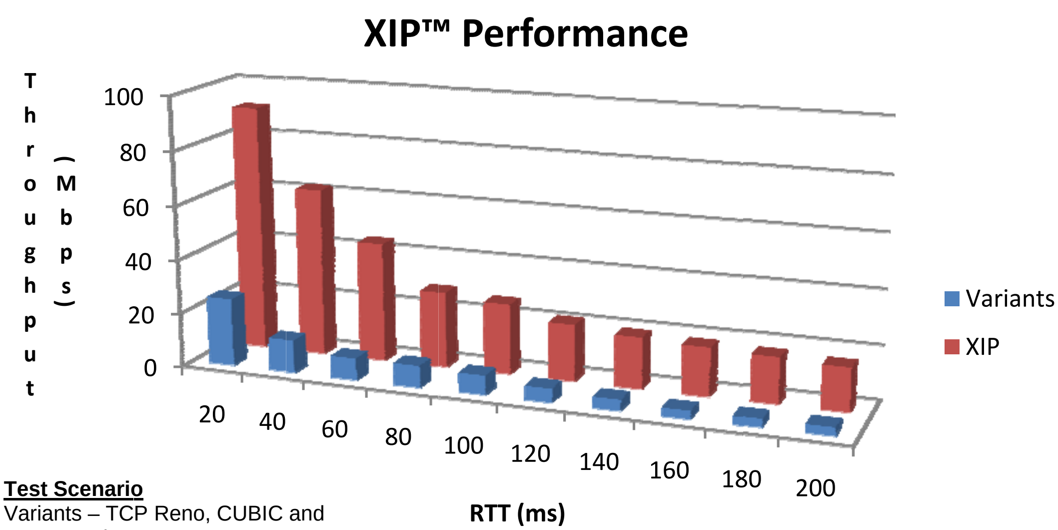
The key to improving application performance and allowing for better utilization of the network's resources is to adjust the limiting parameters of TCP while retaining the facets that still provide a benefit. There is an obvious need for a non-disruptive overlay or enhancement that places no additional burden on either the sending application or the receiving client: it must simply present a better way to facilitate the transfer phase of TCP.

Internap's Accelerated IP or XIP™ is just that solution. XIP is a protocol-compliant service that makes better use of the intelligence resident in the processing of packets and acknowledgements and in turn provides for increased utilization of available bandwidth. The result is significantly accelerated delivery of data between the application and the client because the 'slow downs' and 'slow starts' are minimized in frequency and tempered in behavior.

XIP's first improvement is to reset the size of the client's receiving window or RWND (if necessary). This setting is often arbitrarily configured (too small) and does not synchronize with the capabilities of the network link. Establishing the optimal RWND allows for a greater initial and sustained rate of transmission throughout the session.

Secondly, XIP improves on TCP's approach to Slow Start by cycling it through the ramp up process in an accelerated fashion. The service will request and buffer outbound segments from the sending application by providing ACKs on behalf of the client. This procedural change allows the sending application to accelerate quickly and submit more segments in a shorter period of time. XIP will then receive the ACKs from the client and monitor and record the rate at which they arrive. This intelligence is then used to match future delivery of segments, from its buffer, to that measured rate. XIP is essentially maximizing the initial rate of transmission and maintaining a higher rate of transmission throughout the session.

The final improvement that XIP makes relative to native TCP is in its use of information in assessing the congestion level of the network. As opposed to simply attributing all packet loss to congestion and dramatically slowing the rate of transmission, XIP closely monitors the RTT of ACKs and uses delay as the metric for detecting material congestion. Since the service is collecting and measuring the ACKs from the receiver, it can leverage this intelligence (i.e. the fluctuations in timing) to adjust the rate of transmitted segments. As the congestion subsides, XIP can ramp the rate of transmission quickly to once again match the bandwidth of the connection.



Test Scenario
Variants – TCP Reno, CUBIC and Compound TCP
Receiver O/S – Windows XP
File Delivered: 100MB Compressed
Packet Loss Rate: 1%
Link: 100Mbps

Figure 2: XIP Performance

XIP's architecture allows it to consistently exceed the performance of native TCP (TCP Reno and other variants) when delivering HTTP or FTP content under a multitude of scenarios. Its improvement over the approach to Slow Start and Congestion Avoidance better matches the capabilities of applications, hosts and bandwidth provisioning in place today. The result is accelerated delivery of content and better utilization of existing resources.

Benefits of XIP

- XIP produces a rate of acceleration that consistently provides a two to four time improvement over that of native TCP. This rate of improvement holds across a mix of fluctuating network characteristics such as increasing congestion (expanding RTT) and high levels of packet loss.
- XIP is provided as a service: there is no hardware to buy, install or maintain. Pricing is a function of usage – paying only for traffic that is accelerated.
- There is no client-side application, player or plug-in required. There are no configuration changes required on either the application (sender) or client (receiver) side.
- There is no geographical or theoretical edge to the rate of acceleration. The improvement is between the application and the client – regardless of location. This minimizes the need to expand or increment data center presences to improve application performance.

About Internap

Internap is a leading Internet products and services company that provides *The Ultimate Online Experience*® by managing, delivering and distributing applications and content with 100 percent performance and reliability. With a geographically dispersed platform of data centers, managed Internet services and a content delivery network, Internap frees its customers to innovate, improve service levels, and lower the cost of IT operations. Thousands of companies trust Internap to help them achieve their Internet business goals. For more information, visit http://www.internap.com.