*Warp*Engine™

WHITE PAPER

Required Technology for Today's Network and Application Environments. Your Business Might Not Survive Without it.

BADU networks

- Improving the way the world connects -

CIO Review 20 MOST PROMISING NETWORKING SOLUTION PROVIDERS - 2017

Executive Summary

In computer networking, packet delay variation (PDV), more commonly referred to as jitter, can be defined as random undesirable delays in consistent packet delivery, and it's now the leading cause of poor network throughput. Network jitter triggers corresponding application jitter, and ultimately business process jitter that can have a devastating impact on employee productivity, customer service and revenue. It's driven by the nature of today's network and application environments, and how they interact with the most widely used network protocols.

Today's streaming video, audio, fast data, IoT, voice, and web applications transmit data in unpredictable bursts, becoming a source of jitter before packets even enter the network. In the multi-tenant virtualized public or private cloud environments that increasingly host them, these applications compete for virtual storage, memory, CPU and network resources. In turn, these virtual resources compete for corresponding physical resources, leading to random scheduling conflicts between VMs, hypervisor packet delays, and extra hops between virtual and physical subnets, creating random delays that compound application jitter with virtualization jitter. Volatile last mile wireless connections most users rely on for application access frequently suffer from RF interference, fading and other issues that add still more jitter. Moreover, VPNs that protect business and personal data have become yet another source of jitter by imposing encryption/decryption delays at each network endpoint. All these factors combine to create unprecedented levels of jitter that will become exponentially greater as live streaming, IoT and similar applications proliferate, the move to cloud accelerates, and 5G rolls out.

The ultimate problem beyond the cumulative random delays outlined above, is that widely used network protocols designed for guaranteed packet delivery such as TCP treat jitter as a sign of congestion. They respond by retransmitting packets and throttling traffic to avoid data

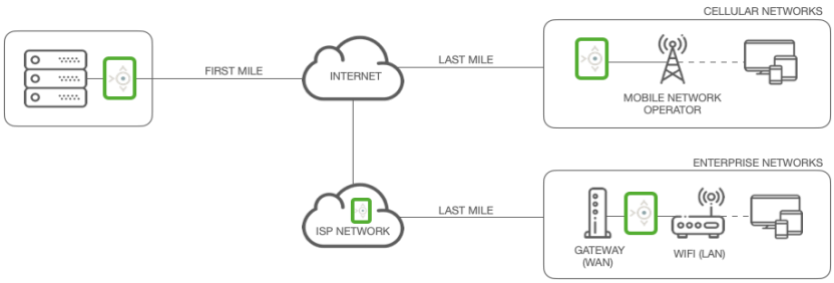
loss until throughput collapses and applications stall. This occurs even when plenty of bandwidth is available, and not only TCP's throughput is impacted.

For cost and operational efficiency reasons, applications using TCP often share the same network and compete for bandwidth with applications using protocols like UDP that sacrifice guaranteed delivery and tolerate packet loss in exchange for maximum speed. This makes UDP and similar protocols a preferred option for many live video and audio streaming, VoIP, and p2p applications, despite the pauses, skips and jumps. Since UDP doesn't react to jitter as TCP does, throughput is almost entirely a function of available bandwidth. Unfortunately, TCP's inability to determine whether jitter is due to congestion results in more bandwidth being allocated to applications using TCP than would otherwise be the case. As a result, bandwidth that could be available to UDP and other traffic is wasted, and the performance of all applications on a shared network suffers.

WarpEngine™ is the only solution that recaptures this wasted bandwidth by eliminating jitter-induced throughput collapse. WarpEngine combines this unique capability with other performance enhancing features that deliver 2x to 10x or greater improvements in throughput for TCP, UDP and other traffic on dedicated WAN, broadband internet, mobile, and Wi-Fi networks. As a result, WarpEngine's breakthrough technology delivers the maximum possible ROI from existing network infrastructure. WarpEngine achieves this at a fraction of the cost and with far superior results to the solution most IT organizations turn to – network upgrades. Although upgrades increase bandwidth to allow more traffic through, they often result in a corresponding increase in the incidence of jitter-induced throughput collapse. This means much of the new bandwidth upgrades provide ends up being wasted because they fail to deal with the root of the problem.

WarpEngine is a single-ended transparent network proxy that requires no changes to client or server applications. In addition, there's no client software to install, making WarpEngine much easier to deploy and maintain, especially in distributed environments with remote users. WarpEngine's single-ended architecture also means it can be deployed at any point on a network, ideally closest to the main source of jitter. This is a major differentiator with traditional dual-ended optimization solutions that require control of both ends of the network – an increasingly difficult requirement to meet in the cloud era.

WarpEngine comes in a range of deployment options supporting throughput of up to 400 Gbps, with 200 million concurrent sessions, and over one million new connections per second. WarpEngine can be installed as a hardware appliance in an ISP's core network, or in front of hundreds or thousands of servers in a corporate data center, working in conjunction with ADCs, SD-WANs and other existing infrastructure.



WarpEngine is interoperable with GTP, enabling it to be deployed at cell tower base stations to optimize mobile networks. To boost Wi-Fi network performance, WarpEngine can be installed in front of access points in large public venue or private Wi-Fi networks to boost upload and download throughput by up to 10x. The WarpGateway™ version of WarpEngine is designed to deliver similar benefits for smaller scale Wi-Fi networks in branch offices, small businesses, retail and hospitality. In larger distributed organizations, WarpGateway can be installed in branch offices to eliminate poor local Wi-Fi network throughput, complementing WarpEngine in the corporate data center.

WarpVM™ is an extension of WarpEngine packaged as a VM for deployment entirely in AWS, Microsoft Azure, Google Cloud or any other public or private cloud environment. WarpVM is ideal for a wide range of use cases including: enterprise cloud and hybrid cloud applications; SaaS vendors seeking to boost performance; content delivery networks (CDNs); eCommerce sites requiring fast page loads; and video on demand (VOD) and video streaming services.

WarpEngine can also be distributed as white-label software on an OEM partner's hardware, or as part of their software stack.

To gain a deeper understanding of how WarpEngine maximizes ROI from existing network infrastructure, and succeeds in today's network and application environments where other solutions fail, it's important to start by looking at TCP's reaction to jitter in greater detail.

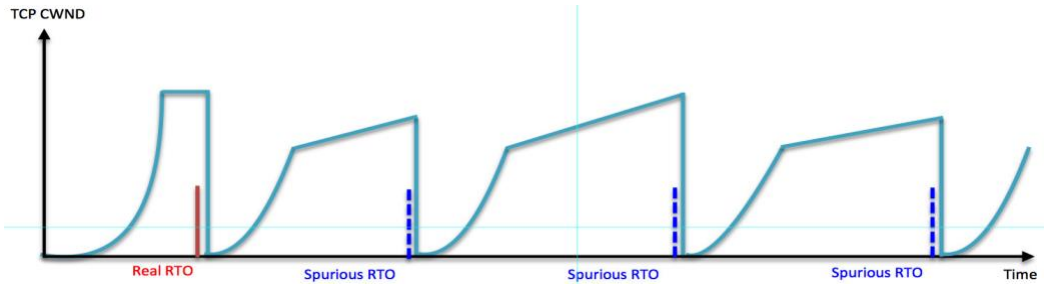
TCP's Reaction to Jitter

When TCP was designed over 30 years ago, the focus was on guaranteeing orderly packet delivery between endpoints. Based on the nature of applications and networks deployed at the time, the underlying assumption was that network packets would arrive in relatively consistent

intervals unless the network became congested.

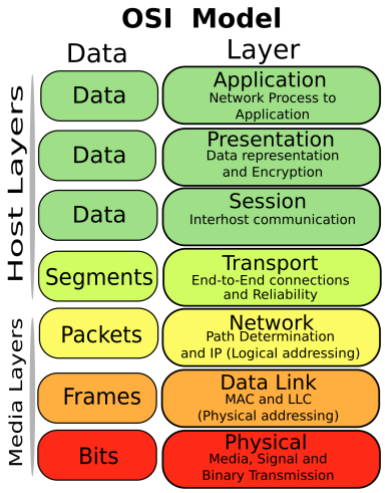
To guarantee packet delivery and guard against data loss, TCP relies on two variables to measure congestion and control throughput: (1) a Retransmission Timeout (RTO) value based on a moving average calculation of the Round-Trip Time (RTT) for each packet to be sent and an acknowledgement (ACK) to be received; and (2) a Congestion Window (CWND) that defines the maximum amount of data that can be transmitted through a connection. TCP sets its retransmission timer to the RTO value as it's sending each packet. If an ACK isn't received before the retransmission timer expires, the packet is flagged as dropped and then retransmitted. With each retransmission attempt the RTO value is increased and CWND is decreased to reduce throughput to avoid further packet loss, on the assumption the network is congested. After three RTOs throughput is halved. After seven RTOs throughput collapses because TCP treats the packets as lost rather than merely delayed, and prevents any packets from being sent. After a sub-second waiting period, TCP begins its recovery process by incrementally increasing CWND with each successful packet transmission, until it reaches its pre-collapse level. This means TCP's recovery process effectively doubles the impact of RTOs by doubling the amount of time available bandwidth is underutilized, and throughput is suboptimal.

Contrary to TCP's original design assumptions, today's streaming applications generate network traffic characterized by periodic bursts of data, causing significant variation in RTT. This is compounded by the virtualized environments they often run in, and the volatile last mile wireless networks users often access them from. RTT variance like PDV, is another term for jitter. It triggers spurious RTOs when ACKs don't arrive from the receiver as expected. Even though plenty of bandwidth is available, TCP responds to spurious RTOs as if the network is congested. It begins the process of reducing throughput until it collapses, and then incrementally recovers to its pre-collapse level, as shown in the image below.



TCP's response to spurious RTOs triggered by jitter wastes bandwidth that could otherwise be allocated to UDP and other traffic. As a result, the performance of all applications on a shared network suffers. While some optimization solutions try to address the symptoms, most don't tackle jitter-induced throughput collapse at all, and only one eliminates the root cause.

Eliminating the Root Cause of Jitter-Induced Throughput Collapse



Like TCP, most optimization solutions pre-date today's jittery network environments and weren't really designed for them, despite what many vendors claim. The Open Systems Interconnection (OSI) model, which divides network functionality into seven layers, provides a useful framework for understanding the benefits and drawbacks of the types of solutions available, and the extent to which they address TCP's response to jitter. They include:

- **Upgrading network bandwidth** – a costly and disruptive physical layer one approach that allows more traffic through, but often results in a corresponding increase in the incidence of jitter-induced throughput collapse, making it counter-productive

- **Multiprotocol Label Switching (MPLS**) – introduced in the 1990s, MPLS operates at layers two and three, the data link and network layers responsible for routing packets to a destination. MPLS ensures performance by using packet labels to force high priority application traffic over pre-determined paths with guaranteed bandwidth allocations. MPLS also aids performance by eliminating the overhead of routing table lookups and independent forwarding decisions at each network hop. Despite these benefits, MPLS doesn't focus on jitter, and it's intended for dedicated WAN links, so its packet labels have no meaning on the internet

- **WAN optimization** – these tools operate primarily at layer seven, the application layer. Introduced around the same time and often used in conjunction with MPLS, they rely on caching, deduplication, and compression to accelerate traffic by reducing its volume. These techniques require payload access, which wasn't a problem initially. However, now that more than 80% of all network traffic is encrypted, it has become a major drawback because it introduces the performance overhead of encryption/decryption delays at each endpoint – another source of jitter. Payload access also imposes the administrative overhead and security risk of exposing sensitive encryption keys to third party vendor tools, which public web sites will never provide. This highlights another significant problem with these solutions – their dual-ended architecture assumes you always have control of both ends of the network, which isn't realistic in today's environment

- **SD-WAN** – a more recent addition to the ranks of network performance solutions, SD-WAN makes it possible to offload branch office internet and cloud-bound traffic from leased line and MPLS links to less expensive broadband. The architectural and cost advantages are obvious. However, there's also a widespread assumption that SD-WAN can optimize performance merely by choosing the best available path among broadband, LTE, 5G, MPLS, Wi-Fi or any other available link. The problem is SD-WAN

makes decisions based on measurements at the edge, but has no control beyond it. What if all paths are bad? Some SD-WAN vendors bundle in WAN optimization solutions, but they come with all the drawbacks noted above. SD-WAN vendors also typically incorporate traffic shaping features that reserve bandwidth for high priority applications, by throttling low priority application traffic. However, traffic shaping doesn't optimize bandwidth for high priority applications by preventing jitter-induced throughput collapse. In addition, it can't be an end-to-end solution if there isn't control over both ends of the network

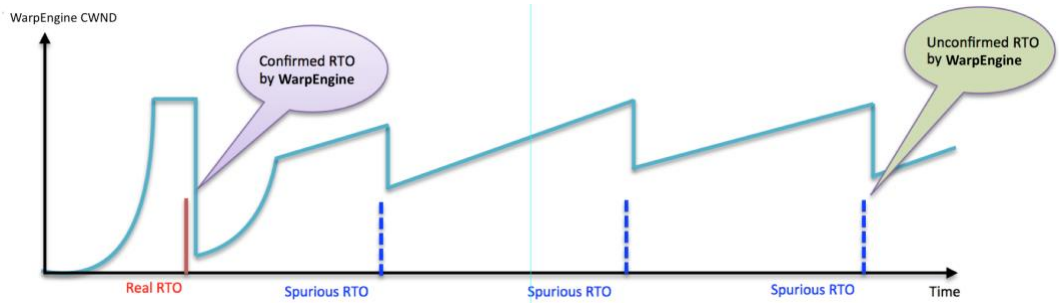
- **Jitter buffers** –order and evenly space packets to realign packet timing for consistency before they're passed to the receiver. Jitter buffers may work for some applications, but they can destroy performance for real-time applications like live video and audio streaming, and create random delays that add to jitter

- **TCP Optimization** – to successfully attack the root cause of jitter-induced throughput collapse, the focus must be on layer four, the transport layer. This is where TCP's congestion control algorithms become a bottleneck by reducing throughput in reaction to jitter having nothing to do with congestion. Some TCP optimization solutions try to address this bottleneck by managing the size of TCP's congestion window (CWND) to let more traffic through a connection, using selective ACKs that notify the sender of which packets need to be retransmitted, adjusting idle timeouts and tweaking a few other parameters. While these techniques can offer some modest improvement, generally in the range of 10% - 15%, they don't eliminate jitter-induced throughput collapse, the resulting waste of bandwidth, or its impact on UDP and other traffic.

Jitter-induced throughput collapse can only be resolved in the transport layer by modifying or replacing TCP's congestion control algorithms to remove the bottleneck they create. However, to be acceptable and scale in a production environment, a viable solution can't require changes to the TCP stack itself, or any client or server applications. It will also have to co-exist with ADCs, SD-WANs, VPNs and other network infrastructure.

Only WarpEngine Eliminates the Root Cause of Jitter-Induced Throughput Collapse

Only Badu Networks' WarpEngine addresses the key requirements outlined above for eliminating the root cause of jitter-induced throughput collapse. In addition, WarpEngine provides other enhancements beyond removing the bottleneck created by TCP's response to jitter, that further improve throughput for TCP, UDP and all traffic on shared networks. Implemented as a transparent proxy, WarpEngine's proprietary algorithms determine if bandwidth is available to a TCP session in real-time. The impact of transient fluctuations in RTT (jitter) and packet loss that trigger RTOs having nothing to do with congestion are filtered out.



As shown in the image above, if an RTO occurs due to actual congestion, WarpEngine's congestion control behaves as TCP's would to protect against packet loss. However, spurious RTOs caused by jitter rather than actual congestion don't result in throughput collapse, the extended recovery period of reduced throughput, and the loss of available bandwidth that normally follows. As a result, WarpEngine recaptures bandwidth that would otherwise be wasted.

This means less bandwidth can be allocated for applications using TCP, since jitter-induced throughput collapse is eliminated and TCP's traffic is optimized. Bandwidth for UDP and other traffic is freed up without negatively impacting TCP's performance. WarpEngine can also be configured to prioritize UDP and other non-TCP traffic over shared networks to enhance their throughput even further without impacting optimized TCP traffic.

WarpEngine's algorithmic approach to optimization offers another key advantage now that over 80% of internet traffic is encrypted. Since WarpEngine doesn't rely on compression and deduplication which require payload access, it eliminates encryption/ decryption delays at each endpoint that increase jitter, as well as the maintenance overhead and security risk of exposing sensitive keys to a third-party solution. Performance and throughput stay at consistently high levels for all types of traffic – encrypted, unencrypted, or compressed.

WarpEngine Architecture

WarpEngine's single-ended proxy architecture enables it to be installed at any point on the network, ideally closest to the main source of jitter. TCP streams arriving at the proxy are terminated, allowing WarpEngine to take over TCP congestion control and provide other performance enhancements without modifying the TCP stack on the client or server. UDP and other non-TCP sessions are not terminated at the proxy. However, WarpEngine can be configured to prioritize UDP and other non-TCP sessions, and enable them to take advantage of other throughput boosting features described below.

WarpEngine consists of two components that work hand in hand to prevent jitter-induced throughput collapse, optimize the use of all available bandwidth, and maximize performance for all traffic over shared networks regardless of protocol:

- A de-bottleneck module that implements WarpEngine's proprietary algorithms which determine if jitter is due to congestion based on bandwidth available to each TCP session in real time, and prevent TCP from reducing the size of CWND and unnecessarily throttling throughput if it's not

- A Transparent Proxy that implements TCP session splicing by splitting the connection between the server and the client into two independent sessions. Each spliced server-to-client TCP session is replaced by a server-to-proxy sub-session and a proxy-to-client sub-session. The two sub-sessions have independent sequence numbers and ACK flows. WarpEngine retains the IP addresses and port numbers associated with the original TCP session source and destination to map them to the new sub-sessions.

This session independence enables WarpEngine to implement its own flow control algorithms based on speed matching that are far superior to TCP's. With speed matching, the proxy receives as many packets as possible, as fast as possible, buffers them, and then forwards them to their destination at different speeds and times. Speed-matching enables another performance enhancing feature - opportunistic bursting. Opportunistic bursting allows WarpEngine to fill unused gaps in bandwidth with TCP as well as UDP and other non-TCP packets that would otherwise be stalled. These capabilities are implemented in a multi-core, multi-threaded architecture capable of supporting throughput of over 400 Gbps, with 200 million parallel sessions, and over one million new connections per second in its maximum configuration. This level of scalability enables use cases beyond the reach of any other solution.

Finally, although WarpEngine is transparent to users, it dramatically improves one of the most visible aspects of user experience - page load times. Browsers only support establishment of two to four TCP sessions simultaneously, whereas a web page can easily have over 100 objects, each requiring its own TCP session to send and receive data. Since WarpEngine connections with the browser are independent of the server and the traffic is optimized, the client will send the next HTTP request much sooner than it normally would. As a result, web pages typically load 2x-3x faster. To put the significance of this improvement in perspective, Amazon calculated that a page load slowdown of just one second costs them $1.6 billion in sales each year.

WarpEngine Provides the Bridge to 5G

RF interference, fading and channel access conflicts frequently cause RTOs leading to jitter-induced throughput collapse in today's 4G LTE networks. In the busy high-speed, small cell, low RTT networks planned for 5G, the impact of RTOs will be devastating, resulting in more jitter-induced throughput collapse than ever, despite the dramatic increase in bandwidth. In addition, LTE networks will serve as the backup for 5G, just as 3G networks have done for LTE. The failover will need to be as seamless as possible, without too dramatic a degradation in performance.

Forward-looking MNOs recognize these issues and test WarpEngine on both their existing 4G LTE and prototype 5G networks. Their results typically show WarpEngine improving existing LTE network throughput by up to 3x or more, and similar improvements for 5G. These results indicate WarpEngine can enable MNOs to control the timing and cost of 5G rollouts by:

- Dramatically improving service and maximizing ROI from existing LTE networks that will provide failover for 5G

- Potentially reducing the number of 5G cells MNOs need to deploy by recapturing bandwidth that would otherwise be wasted due to jitter-induced throughput collapse.

Conclusion

Random delays that create jitter filter up from the transport layer into the application layer, and finally into the business process layer where they impact not only throughput and application performance, but also employee productivity, customer service, and revenue. The MPLS, SD-WAN, WAN optimization, traffic shaping and other solutions that traditionally support network performance operate primarily in the application, network and data link layers. They do an effective job of prioritizing traffic, preventing packet loss, and reducing latency in some instances, so they have a role to play. However, by not focusing on the transport layer they have become much less effective due to the increasingly jitter-prone nature of today's application traffic, the fact that the bulk of it is encrypted or compressed, and travels at least partially over volatile wireless links. The rapid and continuing move to the cloud, the proliferation of IoT, fast data and other streaming applications that transmit packets in random unpredictable bursts, combined with the rollout of busy high-speed, small cell 5G networks will only intensify the impact of jitter, and increase the incidence of jitter-induced throughput collapse in the years ahead. WarpEngine is the only solution capable of eliminating jitter-induced throughput collapse in today's network environments, and future-proofing them against ever-increasing jitter. Your business might not survive without it. See it in action now!

BADU networks
ⓟ 949-310-5390
ⓕ 888-958-7697
ⓔ info@badunetworks.com
2640 Main Street Irvine CA 92614
http://www.BADUnetworks.com