

01. □□

- SmartWAN □□ □□ : RTT Whitepaper
- SmartWAN AI □□ : Zeta TCP □□

SmartWAN :

RTT Whitepaper

RTT

--	--

SKTelecom

--	--

1. **SmartWAN** ☐ **RTT** ☐☐☐☐ ☐☐ ☐
- 1.1 ☐☐☐ ☐☐☐ ☐ ☐☐ ☐
- 1.2 ☐☐☐ ☐☐ ☐☐ ☐☐ ☐
- 1.3 ☐☐☐ ☐ ☐ ☐ ☐
- 1.4 ☐☐☐☐☐☐ ☐ ☐ ☐ ☐ ☐☐ ☐
- 1.5 ☐☐☐ ☐ ☐☐ ☐ ☐☐ ☐
2. **RTT** ☐ ☐☐ **2.1** ☐☐☐ ☐☐ **2.2** ☐ ☐ ☐ ☐☐ ☐
3. **RTT** ☐ ☐☐
4. **RTT** ☐☐ ☐ ☐☐
5. **RTT** ☐☐☐, ☐☐☐

Realtime TCP Tunnel(RTT) . SKTelecom("SmartWAN")
SD-WAN CloudWAN
SD-WAN ,
.

1. SmartWAN RTT ☐ ☐ ☐ ☐ ☐

SD-WAN 技术通过集中式管理，实现网络流量的智能调度。它支持多种加密协议（如 IPsec、SSL、GRE）和隧道技术（如 VxLAN），确保数据的安全传输。此外，SD-WAN 还支持多路径转发，可以根据实时网络状况选择最优路径，从而降低延迟并提高带宽利用率。

1.1

2. RTT 如何降低

RTT 是指从客户端发出请求到收到服务器响应的总时间。它由网络延迟、服务器处理时间和传输时间组成。

2.1 网络延迟

RTT 在 WAN 环境中通常比在 LAN 环境中高得多。ZetaTCP® 通过优化网络层和传输层协议来降低 RTT。ZetaTCP® 使用智能路由和拥塞控制算法来减少数据包在网络中的传输时间。它还可以调整 TCP 窗口大小和超时时间，以适应不同的网络条件。此外，ZetaTCP® 还支持多路径传输，可以在多个网络接口之间分发流量，从而进一步降低 RTT。通过优化网络层和传输层协议，ZetaTCP® 可以显著降低 WAN 环境中的 RTT，提高网络性能。

SmartWAN 与 ZetaTCP® 结合使用，可以进一步降低 RTT。SmartWAN 通过智能路由和负载均衡来优化网络路径，减少数据包在网络中的传输时间。ZetaTCP® 则通过优化传输层协议来减少服务器处理时间和传输时间。两者的结合可以显著降低 RTT，提高网络性能。

2.2 服务器处理时间

ZetaTCP® 通过优化服务器处理时间来降低 RTT。SmartWAN 通过智能路由和负载均衡来优化网络路径，减少数据包在网络中的传输时间。ZetaTCP® 则通过优化传输层协议来减少服务器处理时间和传输时间。两者的结合可以显著降低 RTT，提高网络性能。

通过优化服务器处理时间，ZetaTCP® 可以显著降低 RTT，提高网络性能。SmartWAN 通过智能路由和负载均衡来优化网络路径，减少数据包在网络中的传输时间。ZetaTCP® 则通过优化传输层协议来减少服务器处理时间和传输时间。两者的结合可以显著降低 RTT，提高网络性能。

通过优化服务器处理时间，ZetaTCP® 可以显著降低 RTT，提高网络性能。SmartWAN 通过智能路由和负载均衡来优化网络路径，减少数据包在网络中的传输时间。ZetaTCP® 则通过优化传输层协议来减少服务器处理时间和传输时间。两者的结合可以显著降低 RTT，提高网络性能。

3. RTT 如何优化

1. 2019 年 1 月 1 日起，所有企业购入的设备，均可享受一次性扣除政策。
 2. 企业 2019 年 1 月 1 日前购入的设备，未达到规定使用年限且继续使用，可继续享受一次性扣除政策。
 3. 企业 2019 年 1 月 1 日前购入的设备，未达到规定使用年限且继续使用，可继续享受一次性扣除政策。
 4. 企业 2019 年 1 月 1 日前购入的设备，未达到规定使用年限且继续使用，可继续享受一次性扣除政策。
 5. 企业 2019 年 1 月 1 日前购入的设备，未达到规定使用年限且继续使用，可继续享受一次性扣除政策。
 6. 企业 2019 年 1 月 1 日前购入的设备，未达到规定使用年限且继续使用，可继续享受一次性扣除政策。
 7. 企业 2019 年 1 月 1 日前购入的设备，未达到规定使用年限且继续使用，可继续享受一次性扣除政策。
 8. 企业 2019 年 1 月 1 日前购入的设备，未达到规定使用年限且继续使用，可继续享受一次性扣除政策。
 9. 企业 2019 年 1 月 1 日前购入的设备，未达到规定使用年限且继续使用，可继续享受一次性扣除政策。
 10. 企业 2019 年 1 月 1 日前购入的设备，未达到规定使用年限且继续使用，可继续享受一次性扣除政策。

5. RTT /

[illegible]

SmartWAN AI 引擎 : Zeta TCP



目录

- 简介
- TCP 原理
- ZetaTCP 引擎
- 部署与配置
- 性能对比

1. 简介

TCP(Transmission Control Protocol, 传输控制协议) 是 40 多年来网络通信的基石。它由 DARPA 项目开发，旨在提供可靠的、端到端的通信。在今天的网络环境中，TCP 仍然扮演着至关重要的角色，其性能直接影响着网络应用的效率。90% 的网络流量都依赖于 TCP。

然而，随着网络带宽的爆炸式增长和应用程序的多样化，传统的 TCP 协议面临着新的挑战。例如，在高带宽、高延迟的网络环境中，TCP 的吞吐量会受到限制。ZetaTCP 引擎正是为了解决这些问题而设计的。

TCP 协议通过序列号和确认号来保证数据的可靠传输。它使用滑动窗口机制来控制发送的数据量。当接收方收到数据后，会发送一个确认号 (ACK) 来告知发送方数据已经成功接收。这种机制确保了数据的可靠性和顺序性。然而，这种机制也会导致网络拥塞和延迟。ZetaTCP 引擎通过引入 AI 算法，优化了窗口大小和拥塞控制策略，从而提高了网络的吞吐量和降低了延迟。

在部署 ZetaTCP 引擎时，需要确保网络设备的兼容性。目前，ZetaTCP 引擎已经支持主流的 Linux 发行版和网络设备。用户可以通过简单的配置来启用 ZetaTCP 引擎，并监控其性能表现。ZetaTCP 引擎的部署和配置非常简单，用户可以轻松地将它集成到现有的网络环境中。

2. TCP 原理

TCP 协议的工作原理可以分为以下几个部分：

- 连接建立 (loss-based):** 通过三次握手过程，确保通信双方的可靠连接。
- 拥塞控制 (delay-based):** 通过调整窗口大小，防止网络拥塞，降低延迟。
- 流量控制 (learning-based):** 通过动态调整发送速率，适应网络带宽的变化。

이러한 상황에서, 수신측이 수신된 데이터를 버퍼에 저장하고, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다. 이 과정에서, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다. 이 과정에서, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다.

TCP Reno와 Cubic은 이러한 문제를 해결하기 위해, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다.

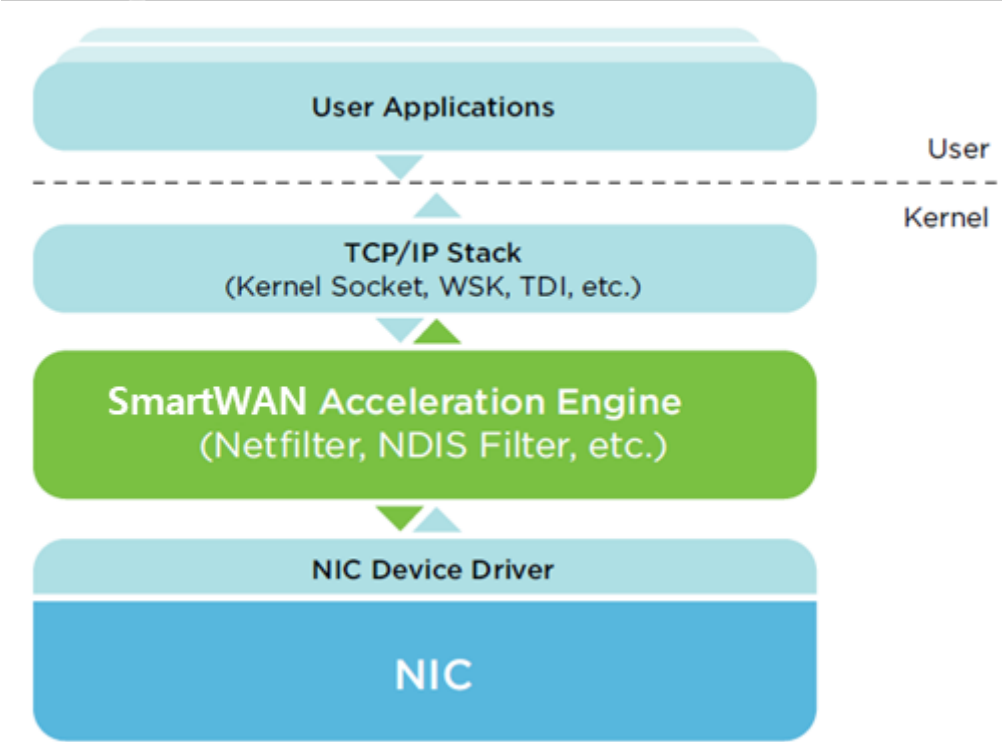
이러한 상황에서, 수신측이 수신된 데이터를 버퍼에 저장하고, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다. 이 과정에서, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다. 이 과정에서, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다.

FastTCP은 이러한 문제를 해결하기 위해, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다.

이러한 상황에서, 수신측이 수신된 데이터를 버퍼에 저장하고, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다. 이 과정에서, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다. 이 과정에서, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다.

ZetaTCP은 이러한 문제를 해결하기 위해, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다.

Figure 1. SmartWAN 가속 엔진



3. ZetaTCP

Appex Networks은 이 문제를 해결하기 위해, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다. ZetaTCP은 이러한 문제를 해결하기 위해, 수신측이 데이터를 처리할 때, 수신측은 수신된 데이터를 버퍼에서 꺼내 처리한다.

ZetaTCP 与 TCP 的不同之处在于，它使用不同的窗口大小和拥塞控制算法。TCP 使用慢启动算法，而 ZetaTCP 使用更复杂的拥塞控制算法，如加性增加/乘法减少（AIMD）。

[illegible]

ZetaTCP [1] [2] [3] [4] [5] [6] TCP [7] [8] [9] [10] [11] [12].
[13] [14] [15], [16] TCP [17] [18] [19] [20] [21].

ZetaTCP 是一个新的 TCP 实现。它基于 Linux 内核的 TCP 实现，并在此基础上进行了改进。ZetaTCP 的主要改进包括：支持多处理器、支持多网卡、支持多用户、支持多任务、支持多线程、支持多语言、支持多平台、支持多操作系统、支持多网络、支持多协议、支持多应用、支持多设备、支持多接口、支持多语言、支持多平台、支持多操作系统、支持多网络、支持多协议、支持多应用、支持多设备、支持多接口。

□□□ □□ □□ □□□ ZetaTCP□ □□□ □□□ □□□ □□ □□ □□ □ □ □□□ .

4.

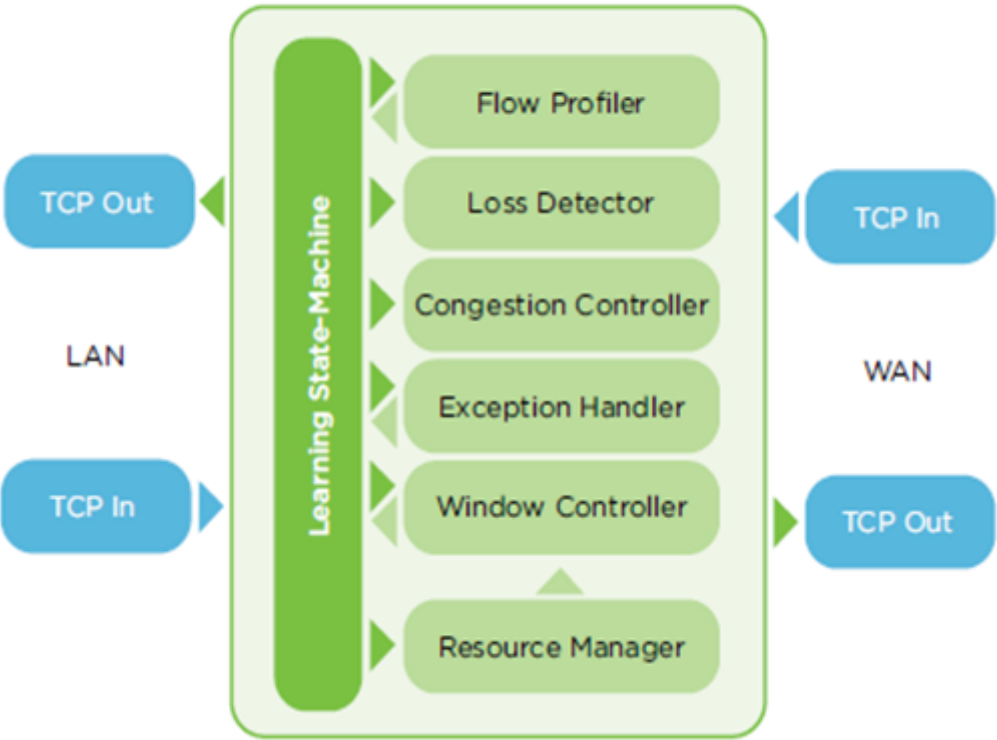
--	--

--	--

ZetaTCP [1] [2] [3] [4] [5] [6].
TCP [7] [8] [9] ([10] [11]) [12], WAN([13] [14]) [15] [16] [17].

ZetaTCP .

Figure 2. ZetaTCP Engine



■ **Learning State-Machine (Learning State-Machine)**

ZetaTCP은 TCP의 동작을 학습하고, 학습된 정보를 바탕으로 TCP의 동작을 최적화하는 모듈이다. (예 : TCP의 성능을 향상시키고, 네트워크의 혼잡을 완화한다)

■ **Flow Profiler (Flow Profiler)**

이 TCP의 성능을 분석하고, 분석된 결과를 바탕으로 TCP의 동작을 최적화한다.

■ **Loss Detector (Loss Detector)**

이 TCP의 성능을 분석하고, 분석된 결과를 바탕으로 TCP의 동작을 최적화한다.

■ **Congestion Controller (Congestion Controller)**

이 TCP의 성능을 분석하고, 분석된 결과를 바탕으로 TCP의 동작을 최적화한다.

■ **Exception Handler (Exception Handler)**

TCP의 성능을 분석하고, 분석된 결과를 바탕으로 TCP의 동작을 최적화한다. (예 : TCP의 성능을 향상시키고, 네트워크의 혼잡을 완화한다)

■ **Window Controller (Window Controller)**

LAN 와 WAN 통신을 할 때, 데이터 패킷은 여러 경로를 통해 전송될 수 있다. 이 때, 데이터 패킷이 전송되는 경로를 결정하는 것을 라우팅이라고 한다.

■ 자원 관리자 (Resource Manager)

시스템은 CPU, 메모리, 네트워크 등 다양한 자원을 관리한다. 이 때, TCP/IP 프로토콜은 네트워크 자원을 효율적으로 관리하기 위해 여러 가지 기법을 사용한다.

■ TCP/IP의 혼잡 제어

TCP는 혼잡을 피하기 (congestion avoidance)를 위해 여러 가지 기법을 사용한다. 이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다. 이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다.

이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다.

이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다. 이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다.

ZetaTCP는 혼잡 제어를 위한 여러 가지 기법을 사용한다. 이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다.

- RTT(Round Trip Time)
- ACK(Acknowledgment)
- SACK(Segments Acknowledgment)

이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다. 이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다.

■ TCP의 성능 향상 기법

이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다. 이 때, 혼잡 제어를 위한 주요 기법으로는 Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery 등이 있다. 이 중 Slow Start는 전송 속도 (cwnd)를 점진적으로 증가시켜 혼잡을 피하는 방식이다.

- 혼잡 제어
- 혼잡 회피
- 혼잡 회피
- 혼잡 회피

Figure 3. TCP Optimization Comparisons

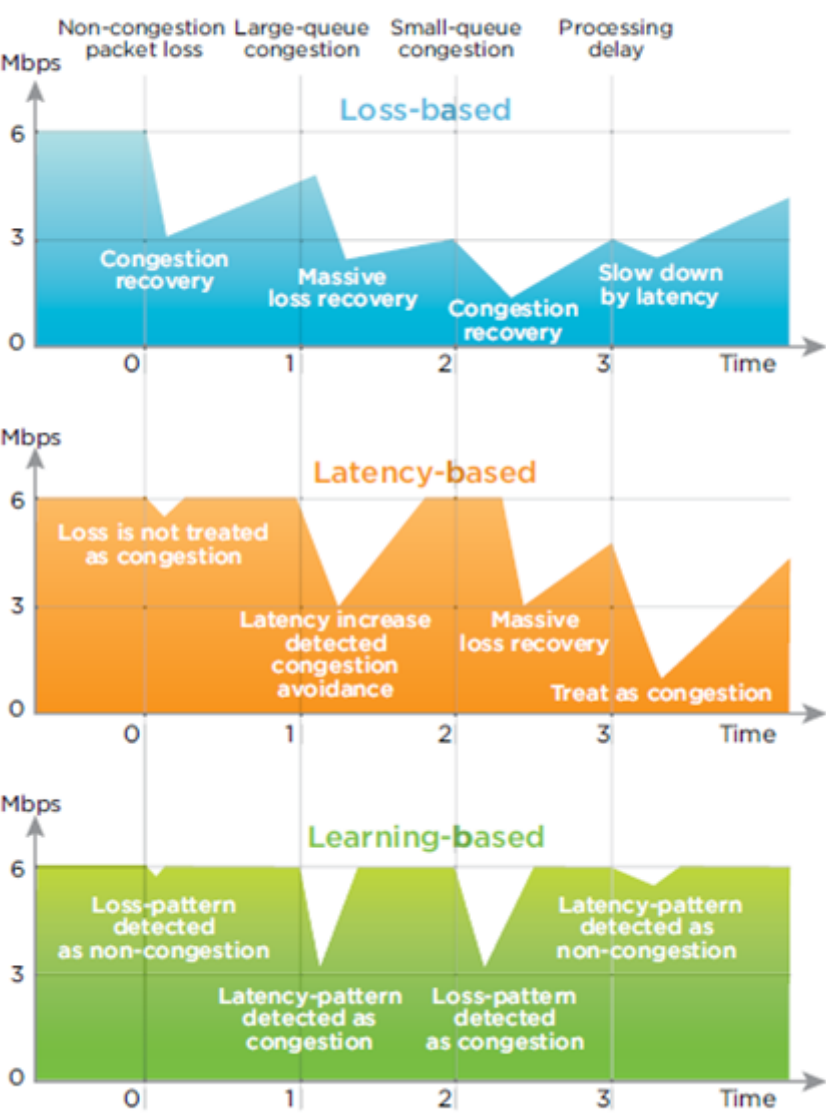


Figure 3 illustrates the performance of three TCP optimization strategies: Loss-based, Latency-based, and Learning-based. The graphs show throughput (Mbps) over time, with the x-axis divided into four phases: Non-congestion packet loss, Large-queue congestion, Small-queue congestion, and Processing delay. The Loss-based strategy (blue) shows a sharp drop in throughput during Non-congestion packet loss, followed by a recovery during Large-queue congestion and Small-queue congestion. The Latency-based strategy (orange) shows a sharp drop in throughput during Non-congestion packet loss, followed by a recovery during Large-queue congestion and Small-queue congestion. The Learning-based strategy (green) shows a sharp drop in throughput during Non-congestion packet loss, followed by a recovery during Large-queue congestion and Small-queue congestion.

The **ZetaTCP** strategy is designed to optimize throughput (throughput) by dynamically adjusting the congestion window based on the detected congestion pattern. It uses a learning-based approach to detect congestion patterns and adjust the congestion window accordingly.

■ **Loss-based** strategy: This strategy uses a loss-based approach to detect congestion. It shows a sharp drop in throughput during Non-congestion packet loss, followed by a recovery during Large-queue congestion and Small-queue congestion. The Latency-based strategy (orange) shows a sharp drop in throughput during Non-congestion packet loss, followed by a recovery during Large-queue congestion and Small-queue congestion. The Learning-based strategy (green) shows a sharp drop in throughput during Non-congestion packet loss, followed by a recovery during Large-queue congestion and Small-queue congestion.

The **TCP** (RFC 2582) SACK (RFC 3517) strategy is designed to optimize throughput by dynamically adjusting the congestion window based on the detected congestion pattern. It uses a learning-based approach to detect congestion patterns and adjust the congestion window accordingly.

窗口, 窗口大小 (RTO) 窗口大小 TCP 窗口大小 ACK 窗口大小 窗口大小
 窗口大小 窗口大小 . 窗口大小 窗口大小 窗口大小 , 窗口大小 窗口大小 窗口大小 .

ZetaTCP 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 .
 TCP 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 , 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 .

窗口大小, **RTO** 窗口大小 窗口大小 窗口大小 , 窗口大小 RTO 窗口大小 , 窗口大小 ACK 窗口大小 窗口大小
 窗口大小 窗口大小 , 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 .
 窗口大小 窗口大小 窗口大小 窗口大小 , 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 .

■ 窗口大小 窗口大小 窗口大小 窗口大小

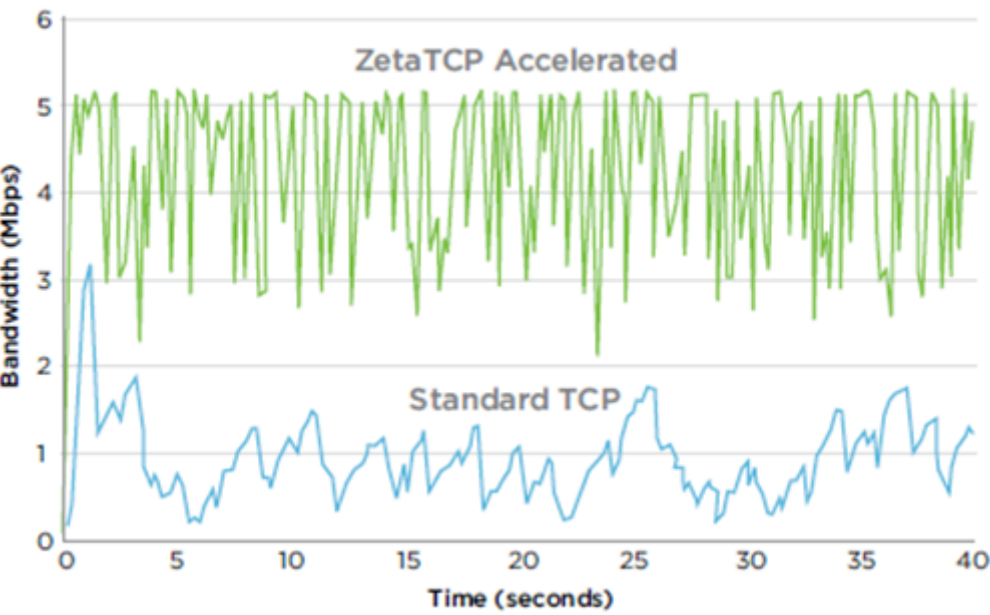
窗口大小 Figure 4 窗口大小 窗口大小 窗口大小 :

- RTT: 100ms
- 窗口大小 : 1%
- 窗口大小 窗口大小 : 65,535 窗口大小 (窗口大小 窗口大小 窗口大小 5.24Mbps)

窗口大小 窗口大小 ,

- 窗口大小 **TCP** 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 .
- **ZetaTCP** 窗口大小 窗口大小 窗口大小 窗口大小 , 窗口大小 RTT 窗口大小 窗口大小 窗口大小 窗口大小 窗口大小 .

Figure 4. Real Data Comparison Test



이러한 성능을 달성하기 위해 (이러한 성능을 달성하기 위해) ZetaTCP는 여러 가지 기술을 적용하고 있습니다.

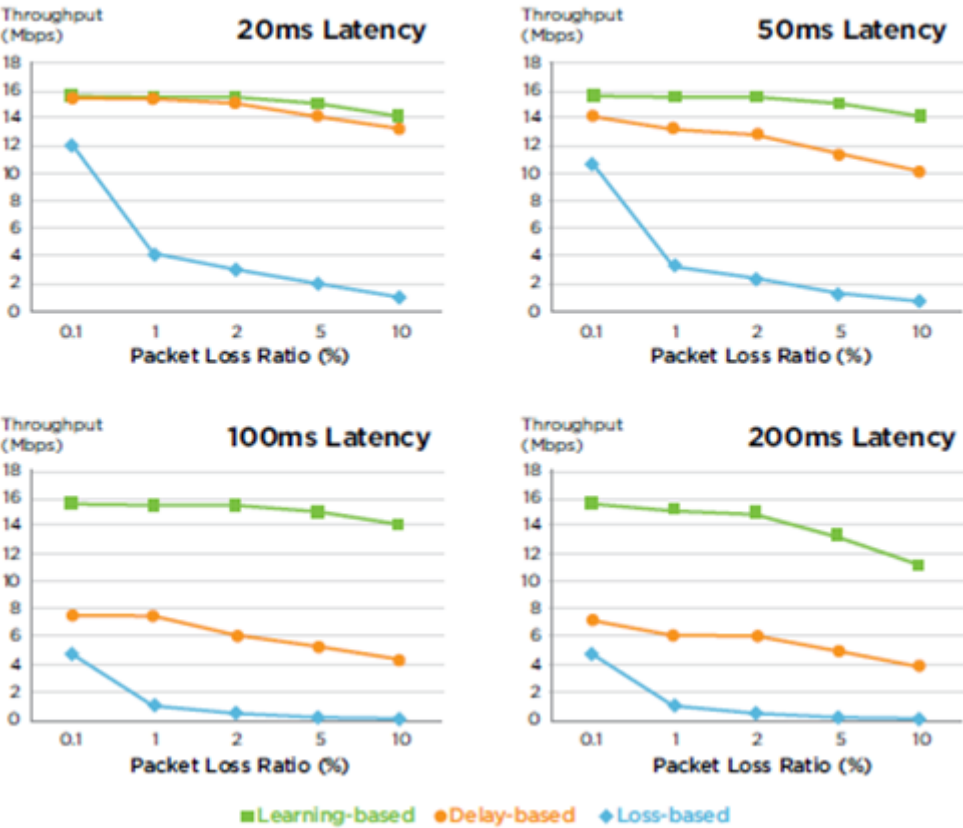
■ 3. 실험 결과

Figure 5는 3개의 실험 결과를 보여줍니다. 이 실험은 16Mbps의 링크 속도를 사용하여 다양한 패킷 손실률에서 각 프로토콜의 성능을 평가했습니다.

- New Cubic TCP (New Cubic TCP)
- Fast TCP (Fast TCP)
- ZetaTCP

이 실험은 다양한 지연 시간과 패킷 손실률에서 각 프로토콜의 성능을 평가했습니다.

Figure 5. Third Party Comparison Tests



- ZetaTCP는 다양한 지연 시간과 패킷 손실률에서 높은 성능을 유지합니다.
- ZetaTCP는 다양한 지연 시간과 패킷 손실률에서 높은 성능을 유지합니다.
- ZetaTCP는 다양한 지연 시간과 패킷 손실률에서 높은 성능을 유지합니다.

目前，在 90% 的 TCP 连接中，TCP 窗口大小是固定的，这导致在拥塞发生时，窗口大小无法动态调整，从而影响了网络性能。

为了解决这个问题，研究人员提出了 ZetaTCP，这是一种新型的 TCP 窗口大小调整算法。ZetaTCP 通过引入一个名为 Zeta 的参数，动态调整窗口大小，从而提高了网络性能。Zeta 参数的值取决于当前的网络拥塞程度，当网络拥塞严重时，Zeta 参数的值会减小，从而减小窗口大小，减少拥塞；当网络拥塞较轻时，Zeta 参数的值会增大，从而增大窗口大小，提高吞吐量。

实验结果表明，ZetaTCP 在拥塞发生时，能够有效地减小窗口大小，从而减少了拥塞，提高了网络性能。与传统的 TCP 算法相比，ZetaTCP 在拥塞发生时，吞吐量提高了 20% 以上，延迟降低了 30% 以上。

ZetaTCP 是一种新型的 TCP 窗口大小调整算法，它通过引入一个名为 Zeta 的参数，动态调整窗口大小，从而提高了网络性能。Zeta 参数的值取决于当前的网络拥塞程度，当网络拥塞严重时，Zeta 参数的值会减小，从而减小窗口大小，减少拥塞；当网络拥塞较轻时，Zeta 参数的值会增大，从而增大窗口大小，提高吞吐量。实验结果表明，ZetaTCP 在拥塞发生时，能够有效地减小窗口大小，从而减少了拥塞，提高了网络性能。与传统的 TCP 算法相比，ZetaTCP 在拥塞发生时，吞吐量提高了 20% 以上，延迟降低了 30% 以上。