

# 镭速传输协议 技术白皮书

深圳市云语科技有限公司

## 文档信息

<b>项目：</b>	镭速传输		
<b>作者：</b>		<b>版本号：</b>	V1.0
<b>标题：</b>	镭速传输协议技术白皮书	<b>版本日期：</b>	2019/4/30
<b>由通过：</b>	名称，组织。单元	<b>批准</b>	

## 版本历史

版	更改说明	作者	日期
1.0	与新模板初始文档		2019年4月 30日
1.1	翻译英语		

## 目录

前言.....	4
1 TCP 在不同时延&丢包网络下的工作情况.....	4
2 拥塞检测控制技术简介.....	4
2.1 Loss-based TCP 改进尝试与局限 .....	5
2.2 Delay-based TCP 改进尝试与局限.....	5
3 传输协议技术改进方向.....	5
3.1 基于 IP 重新定义全新的协议和拥塞控制机制.....	6
3.2 基于 UDP 全新的协议和拥塞控制机制.....	6
4 镭速传输协议主要技术原理.....	6
4.1 更有效的拥塞判断及处理 .....	6
4.2 更准确及时地进行丢包判断恢复 .....	7
5 镭速传输协议对比 TCP 协议测试数据 .....	7
6 镭速传输协议主要特性.....	8
6.1 高速传输 .....	8
6.2 实时性好 .....	8
6.3 防火墙&NAT 设备友好性.....	8
6.4 容易集成 .....	8
6.5 高度可配置 .....	8

6.6 跨平台 .....	8
7 镭速传输协议主要传输技术指标参考 .....	9
7.1 100Mbps 网络带宽下，不同的丢包率/时延传输对比.....	9
7.2 镭速传输协议支持的最大传输带宽 .....	10
8 结论.....	10

# 前言

传输控制协议 TCP (Transport Control Protocol) 是 TCP/IP 协议栈中的传输层协议。根据国际权威组织统计, 目前全球互联网数据流量 90% 以上通过 TCP 传输, 通过其他传输的不足 10%, 并且 TCP 所占份额还在继续扩大。TCP 通过序列确认以及包重发机制, 提供可靠的数据发送; 同时各种广泛使用的操作系统 (WINDOWS/LINUX/UNIX/MAC) 均内置了 TCP 协议栈, POSIX 标准定义了 TCP Socket 标准 API 接口, 这些因素促使 TCP 在全球范围内获得了极广泛的成功。

但是这一设计于二十多年前的传输协议已经越来越不适应飞速发展的网络环境和新型应用的要求, 尤其在网络传输路径上存在一定的丢包和延时的情况下, TCP 协议传输吞吐量急剧下滑, 常常无法有效地利用路径带宽, 导致传输速度慢、传输时间长以及传输体验差。

## 1 TCP 在不同时延&丢包网络下的工作情况

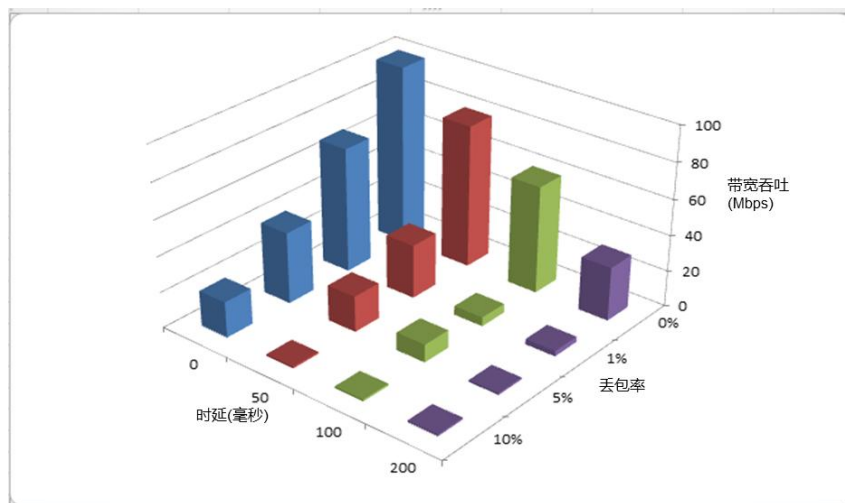


图 1: 时延&丢包对 TCP 传输吞吐影响

从上图可以看到, 随着网络丢包率和时延的变大, TCP 协议的带宽吞吐率急剧下降, 在 10% 丢包情况下, 大于 50 毫秒的时延, 带宽吞吐率不到 1Mbps;

## 2 拥塞检测控制技术简介

一般来说, 传输协议的带宽吞吐能力由拥塞检测、控制技术决定。以拥塞检测处理方式区分, 主要分为两种:

- *Loss-based* (基于丢包的拥塞检测及处理) - 以丢包来检测拥塞并调整传输速度; 该

类拥塞算法的典型代表为 TCP-Reno、TCP-Cubic 算法；

● *Delay-based*（基于延迟的拥塞检测及处理）- 以往返延迟（*Round Trip Time, RTT*）的变化来检测拥塞并调整传输速度； 该类拥塞算法的典型代表为 TCP-Vegas、fastTCP 算法；

## 2.1 Loss-based TCP 改进尝试与局限

*Loss-based TCP* 改进技术以丢包来判断拥塞并调整传输速度的方式。加速改进主要是通过：*Loss-based TCP* 加速改进主要如下方面：

- 增大初始拥塞控制窗口(*Congestion Window, CWND*)；
- 在出现丢包时，使用比传统 *TCP* 更激进的方式恢复 *CWND*，以期减少拥塞对传输速度的影响。

这些改进在部分情况下确实能够提升一定速率，但 *Loss-based TCP* 加速技术在原理上具有以下两个严重问题：

- 1、以丢包作为传输网络路径拥塞发生的信号并不准确，现代很多网络会产生非拥塞因素的丢包，特别是对无线网络，如空口传输误码/无线信号被干扰等因素导致的丢包并不意味着拥塞发生；
- 2、现代网络设备通常缓存队列很深，当拥塞发生时，设备缓存传输队列变长，传输延迟显著提高，但丢包并不会发生。*Loss-based TCP* 加速机制将继续高速传输直到队列完全充满以致溢出，这种情况会导致大批量的数据包丢失。不但加重路径节点拥塞，而且需要花更长时间从大量丢包中恢复过来，经常会导致传输阻滞/传输实时性变差。

## 2.2 Delay-based TCP 改进尝试与局限

*Delay-based TCP* 改进技术克服了 *Loss-based TCP* 的主要缺陷，在原理上采用往返延迟的变化来判断拥塞程度并相应调整传输速度。*Delay-based TCP* 加速技术不将丢包当作拥塞，在非拥塞因素发生的丢包时可以保持较高的速率。因此，设计优良的 *Delay-based TCP* 比 *Loss-based TCP* 在传输速度上有了一定的提升。但是 *Delay-based TCP* 在使用中同样存在以下缺陷：

- 1、如果 *TCP* 连接的路径上节点设备的队列很浅，发生拥塞时往返延迟并不提高，拥塞体现为阵发的丢包。*Delay-based TCP* 技术感知不到这种拥塞，会继续高速发送数据包，从而导致持续大量丢包，之后的丢包恢复期会很长，导致传输速度显著降低甚至传输阻滞。
- 2、当网络路径本身的延迟变化很大时，*Delay-based TCP* 技术会将非拥塞因素导致的延迟增加误判为拥塞并转入拥塞处理，从而导致没有必要的压低传输速度。尤其在当前的移动互联网时代，无线网络延迟变化就很频繁，一些网络设备（特别是安全设备）也会不定时地引入额外的数据包处理延迟。

## 3 传输协议技术改进方向

TCP 在广域网传输中传输效率低下的问题引起了学术界和产业界的关注，从 21 世纪初以来，陆续有研究者和商业公司提出了自己的改进方案，改进方案主要分为两类：

- 1) 基于 IP 重新定义全新的协议和拥塞控制机制;
- 2) 基于 UDP 定义全新的协议和拥塞控制机制;

### 3.1 基于 IP 重新定义全新的协议和拥塞控制机制

SCTP: SCTP 最早是电信组织为了将 TDM 网络下的 SS7 信令移植到 IP 网络上运行而提出的传输协议, SCTP 提出了 Multi-Homing、Multi-Streaming、Heartbeat over path、NACK 等一系列新的提高传输协议能力的特性, 一度作为可能替代 TCP 的一颗耀眼的明星。但 SCTP 经过一段时间的应用后归于沉寂, 其主要原因为:

SCTP 直接基于 IP 协议, 是一个全新的与 TCP/UDP 并行的传输层协议, 全球互联网中有大量的已经部署的 NAT 设备只支持 TCP/UDP 协议, 无法识别 SCTP 协议, 导致 NAT 设备后的计算机无法正常通过 SCTP 协议进行全球互联;

### 3.2 基于 UDP 全新的协议和拥塞控制机制

UDP 协议与 IP 协议提供的服务一致, 仅仅比 IP 协议多 8 个字节的 UDP 头部; UDP 与 TCP 一样, 能够顺利的通过已存的所有 NAT 设备, 因此大量的研究力量投入到基于 UDP 定义全新的协议工作上。

开源产品代表: UDT (UDP-based Data Transfer Protocol)

商业产品代表: 国内的主要厂商有瑞云科技的镭速传输协议; 国外主要厂商有 Asperasoft 的 fasp 传输协议;

## 4 镭速传输协议主要技术原理

镭速传输协议主要通过如下两方面来改善传输效率:

- 更有效的拥塞判断及处理
- 更准确及时地进行丢包判断恢复

### 4.1 更有效的拥塞判断及处理

目前主流的拥塞判断是基于二十几年前的网络情况设计的, 其基本假设是任何丢包都反映了网络拥塞。这一假设与现代网络情况已完全脱节。现代网络丢包常常并不是由拥塞因素引起的。这一脱节导致传输协议常常进入过于保守的传输状态。

镭速传输协议拥塞检测算法会自动收集路径上已有的背景传输信息(丢包、时延以及抖动), 根据传输速度准确判断出实际的拥塞情况, 既不过于保守也不过于激进, 可以有效充分利用路径带宽。

## 4.2 更准确及时地进行丢包判断恢复

标准 TCP 协议栈通过两种手段判断丢包：

- 1、接收端连续重复确认包（Dup-ACK）的数量；
- 2、ACK 超时；

当有较多丢包时，往往要靠 ACK 超时来判读超时并引发重传。现代网络的丢包经常是阵发，一个连接上有多个数据包同时丢失是常有的事。因此标准 TCP 经常要靠超时来重传补洞，往往导致出现几秒甚至上十秒的等待状态，让传输长时间停滞甚至中断。这是影响标准 TCP 效率主要问题之一。

镭速传输协议放弃了 TCP 协议把传输报文序列既用来作为传输的字节计数又作为可靠传输的确认标识的做法，设计了数据计量和包确认分别独立的全新 ACK 算法。传输发送方根据传输接收方返回的 ACK 信息，传输发送方可以第一时间精确判断出丢包情况并进行数据重发，而不用依赖多个 DACK 的累计确认或 ACK 超时定时器来触发数据重发，极大的提升了传输速度和传输实时性。

## 5 镭速传输协议对比 TCP 协议测试数据

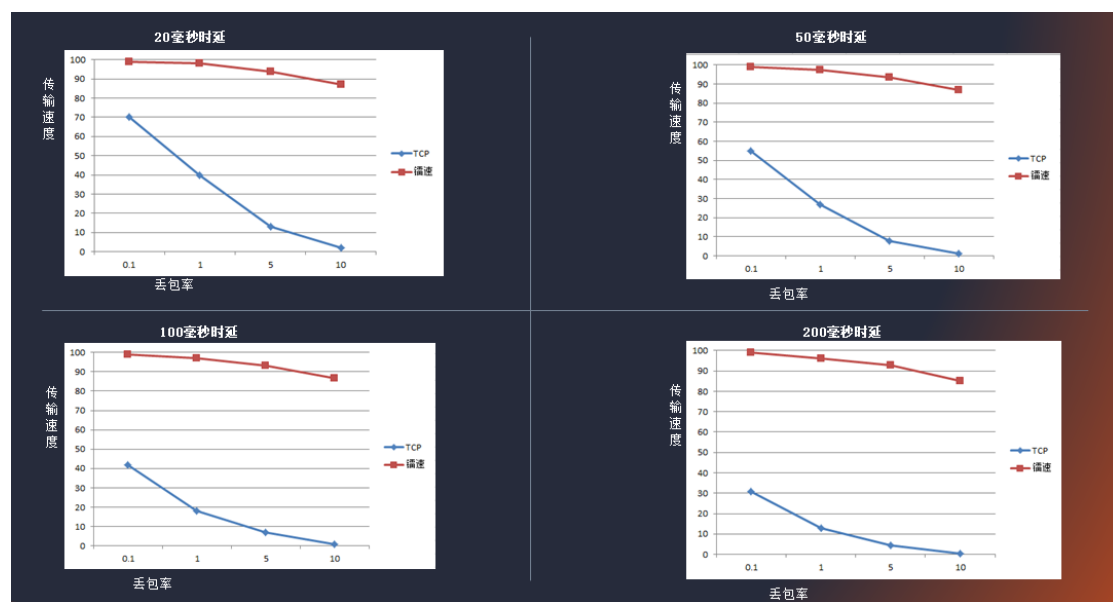


图 2：不同延迟及丢包率对比 TCP

我们测试了从[20 毫秒,50 毫秒,100 毫秒,200 毫秒] 4 种不同的延迟，对比了从[0.1%, 1%, 5%, 10%] 4 种丢包率下，镭速传输协议与 TCP 传输协议的对比。

从图示测试数据可以看到，在各种试验下，随着丢包率的增加，TCP 的传输速度急剧下降，而镭速传输协议可以持续保持在一个合理稳定的传输速度。



## 6 镭速传输协议主要特性

### 6.1 高速传输

镭速传输协议可以充分利用大带宽网络，以最快的速度进行数据传输。在海量数据分发应用场景中，可以帮助用户在最少的时间完成大数据传输。

### 6.2 实时性好

镭速传输协议支持传输连接多通道并行特性，当用户在 A 点与 B 点之间建立连接后，可通过多并行通道特性快速并行传输多个无关的数据流，这些数据流的传输无需经历连接建立的握手过程，极大提高了传输实时性；镭速传输协议创新性的 ACK 设计，帮助用户在存在丢包的环境中，以最快最准确的方式完成数据重发。在游戏、直播等应用场景中帮助用户实现最佳数据传输实时性。

### 6.3 防火墙&NAT 设备友好性

镭速传输协议基于 UDP 协议，可以顺利通过各种 NAT 设备；  
镭速传输协议可以在 1 个 UDP 端口上运行多条连接，镭速传输协议服务端只需要用户的防火墙设备开放 1 个端口即可与多个镭速传输协议客户端完成连接和数据传输。

### 6.4 容易集成

镭速传输协议运行在系统的应用层和用户空间，不需要修改操作系统内核配置；镭速传输协议提供一系列简单易用的 SDK(Software Develop Kit/软件开发套件)、API (Application Programming Interface/应用编程接口)以及清晰完整的开发文档，帮助用户快速集成。

### 6.5 高度可配置

镭速传输协议提供配置参数供用户灵活自定义，通过不同的参数组配置，让传输协议可以最好的适用于用户的应用场景。例如大文件传输更加关注带宽的利用率，游戏/直播等应用更加关注数据传输的实时性，镭速传输已经针对常见的多种应用场景进行优化，提供多组配置参数供用户进行选择。

### 6.6 跨平台

镭速 Proxy 支持 Windows/Mac/Android/IOS/Linux/Unix 等主流计算平台。

## 7 镭速传输协议主要传输技术指标参考

### 7.1 100Mbps 网络带宽下，不同的丢包率/时延传输对比

网络场景	丢包率/时延指标	TCP 传输速度	UDT 传输速度	镭速协议传输速度
局域网	丢包率 0%、时延 1~10ms	11MB/s	11MB/s	11MB/s
省内不同城市	丢包率 1%、时延 60ms	小于 1000KB/s	4.5MB/s	10MB/s
国内不同省份	丢包率 3%、时延 100ms	小于 200KB/s	2.3MB/s	9.5MB/s
不同运营商网络	丢包率 5%、时延 150ms	小于 100KB/s	1.5MB/s	8~9MB/s
跨国网络	丢包率 20%、时延 150ms	无法工作	小于 50KB/S	6MB/s

备注：UDT 是指开源界影响较大、应用较多的基于 UDP 实现的可靠传输协议，参见 <http://udt.sourceforge.net/>

## 7.2 镭速传输协议支持的最大传输带宽

- 单个镭速传输协议进程/线程支持 1Gbps;
- 对于用户传输带宽大于 1Gbps 的场景，可以通过部署负载均衡设备，在负载均衡设备后部署多个镭速传输协议进程/线程，可以支持水平扩展，传输能力无上限；

## 8 结论

广泛使用的标准 TCP 协议已难以适应现代网络的复杂性和多样化需求，瑞云科技镭速传输协议从设计原理上解决了现有 TCP 传输协议的核心问题，可以快速部署在已有的设备上，无需重度投资和重复投资，即可帮助用户快速传输其数据，在数据海量增长爆发的时代里，必将发挥更大的作用。