RTT 实时 TCP 隧道技术说明

北京华夏创新科技有限公司

名词和缩略词

CloudWAN：AppEx 广域网优化 SD-WAN。

POP：Point Of Presence，入网点。

RTT：Real-Time TCP Tunnel，实时 TCP 隧道。隧道包含了两个 POP 之间 1 条或多条 ZetaTCP[1] 加速的 TCP 连接。在 Linux 系统中，RTT 可以作为标准网络接口使用。

Route：在本篇文档中，路由定义了 1 条 RTT 的 POP 路径。RTT 的路由开始于 起始 POP，中间可以经过最多 2 个（最少 0 个）中间 POP（iPOP），最后到达 终结 POP。同一条 RTT 隧道的 2 个方向的路由可以不同，即来去可以不同路，在参数中可以设定是否允许来去不同路。

为什么需要 RTT？

CloudWAN 作为软件定义的广域网，为了能够提供给用户更高效的传输优化服务，需要在 POP 间搭建极速虚拟通道，克服互联网传输瓶颈并消除物理链路丢包。AppEx 为此专门研发了 RTT 技术。

用户在使用 CloudWAN 传输优化服务、尤其是在跨国环境下使用该优化服务时，往往会发现原本物理链路上面存在的丢包情况在使用优化服务后被完全消除或者大幅降低。

CloudWAN 使用 RTT 技术搭建虚拟链路，用户数据是流通在虚拟链路之上。RTT 虚拟链路就像一层铁路，将地面（物理链路）的崎岖不平（丢包）滤掉，给火车（用户流量）提供一条平滑的运行轨道（传输隧道）。

RTT 主要技术

就像铁路需要枕木和铁轨，RTT 依赖两个主要技术实现。

1）ZetaTCP 协议优化技术

RTT 隧道基于 TCP 协议实现，并融合了 ZetaTCP 协议优化算法，使 RTT 隧道的数据传输效率和可靠性得到充分保障。

可靠性：TCP 发送方发出的每一个数据包都要求接收方反馈确认是否收到该包；如因链

路丢包等因素导致接收方没有收到，则发送方会重新发送该包（一次或多次）直到对方确认收到；这种“发送-确认”机制保证了 TCP 传输数据的可靠性。而基于 TCP 实现的 RTT 隧道自身也具备了这种传输可靠性，在传输用户流量时，如果遇到物理链路丢包，则隧道自身载体会重传丢失的包，把链路丢包内部消化掉，用户流量完全感知不到底层物理链路的丢包，因此体验更好。

传输效率：ZetaTCP 协议优化算法让网络数据在相同条件下跑得更快，将此技术应用于 RTT 隧道，就会使隧道的传输效率更高。

2）FEC（Forward Error Correction）前向纠错技术

FEC 通过在传输数据中增加额外的冗余校验码达到在出现丢包时无需重传即可恢复所传输用户数据的效果。简单来说，FEC 在所传输数据包中额外携带一部分冗余数据，一旦出现丢包情况，就可以通过这些冗余数据在接受一侧将丢失的数据加以恢复，避免发送方再次重传。这就像是消费者在网上购买 1 kg 的水果，商家直接发了 1.3 kg 的水果过来；如果中间快递运输环节导致部分水果损坏，消费者就可以通过额外的 0.3 kg 水果自行补偿，避免了商家再重新补发的消耗。

RTT 隧道类型

任何收益都需要付出相应的代价。在 RTT 的机制中，链路丢包的消除是以相应的链路延时的增加为代价的（隧道自身进行丢包补偿重传会导致相应的延时的增加）。

考虑到不同网络业务对链路状况的敏感点不同，可将业务区分为“丢包敏感”型业务和“延时敏感”型业务。其中，几乎所有基于 TCP 协议进行交互的业务均为“丢包敏感”型业务，如 HTTP、FTP 等等；这类业务对延时敏感度不高，一定的延时增加并不会导致其业务响应效率的显著下降，但是只需极少量的丢包就会导致大幅的业务效率下降。而另一些业务如 VOIP、视频会议等实时交互类业务，少量的丢包仅仅会导致一两个音调或画面像素的缺失，并不会导致沟通效果的下降；但是延时的增加则会造成交互实时性的下降。

为充分保障所有业务经 RTT 传输优化后的最佳效果，在隧道算法设计时定义出了两种类型的 RTT 隧道：TCP-RTT 和 Non-TCP-RTT。其中，

TCP-RTT 用于承载“丢包敏感”型业务，通过丢包补偿算法最大限度的保障隧道内无丢包的环境；

Non-TCP-RTT 则用于承载“延时敏感”型业务，通过预设的最大延时阈值确保隧道在进行丢包补偿时不会出现太大的延时增加，当延时增长超过设定阈值时，允许出现可控的丢包以缓解延时增长的幅度。

两类隧道在基础算法实现上完全一致，是通过所提供的参数配置定义出两类隧道类型。

路由优化

RTT 的路由开始于 起始 POP，中间可以经过最多 2 个（最少 0 个）中间 POP（iPOP），最后到达 终结 POP。iPOP 可以帮助实现路由优化。这时 RTT 虚拟链路就像一条铁路线，为火车（用户流量）提供了一条最优的线路，从始发站（起始 POP）出发，途中可以经过由铁总（中央控制器）选定的经停站点（iPOP），最终到达终点站（终结 POP）。如果途中发生突发情况，例如部分线路拥堵等，铁总会随时切换经停站点，保证火车准点到达。而封装好的 RTT 流量在流经 iPOP 时的切换过程就像火车途径道口的铁路扳道岔一样，内层的用户流量不会感知到路线的变化，也不会中断。

为了能够实现动态切换路由的同时 RTT 隧道不中断，我们在 TCP 连接外套了一层 UDP 封装来承载 RTT 隧道。在实际传输中，首先通过起始 POP 系统中运行的路由控制模块（与中央控制台实时通讯）在外层 UDP 封装的包头中写入 iPOP 信息；之后，在流量流经 iPOP 时，只需对最外层的 UDP 封装进行拆封即可获取下一跳 POP（iPOP 或终结 POP）的信息；随后，流量即被送往下一跳目的地。此过程中，内层的 RTT 隧道完全不会中断，以保障最内层用户流量的无中断传输。

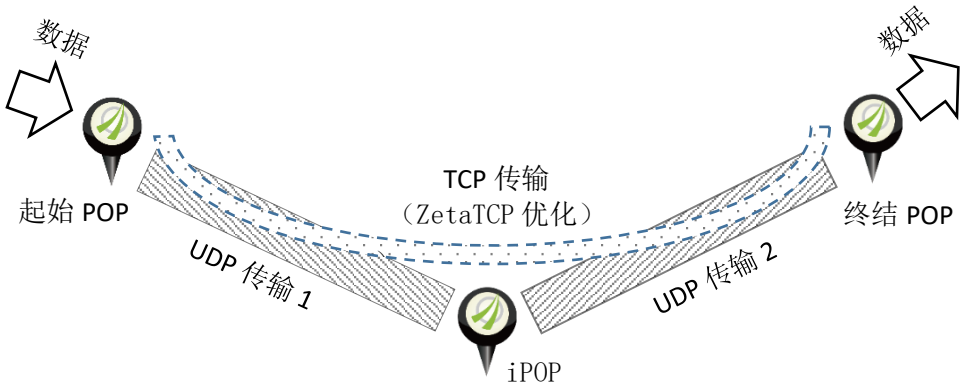

图 1 RTT 隧道路由示意图

如图 1 所示，数据进入 起始 POP，以 ZetaTCP 优化的 TCP 协议进行传输，会被再封装

进 UDP 数据包中，发送给 iPOP，iPOP 中转数据，最终发送给 终结 POP。对于封装在 UDP 中的 TCP 连接，整个链路表现为一条虚拟的点到点链路。

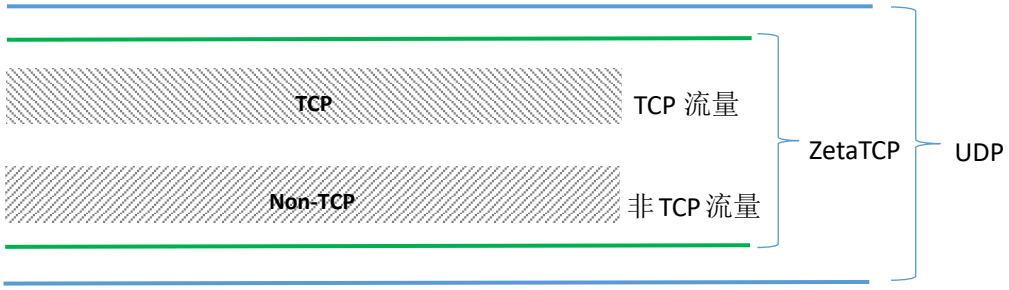

图 2 RTT 隧道内部示意图

参考文献：

[1] ZetaTCP 单边加速技术白皮书

时间改变一切，速度成就未来

北京华夏创新科技有限公司

AppEx Networks Corporation

400-0027-739

北京市海淀区北清路 68 号用友软件园北区 16 号 C 座 6 层