

长距离高带宽环境下 UDT 协议的分析与应用^①

Analysis and Application of UDT Protocol in Long-Distance High-Speed Networks

张磊 葛敬国 李俊 (中国科学院 计算机网络信息中心 北京 100190)

摘要: UDT 是一种面向长距离高带宽网络的 UDP 改进型传输协议。首先探讨了 UDT 协议自身的特点, 然后通过与当前几种主要的 TCP 和 UDP 改进型协议的对比, 以及对网络实际的测量, 提出 UDT 协议在大数据量传输方面更为适合, 最后分析了 UDT 协议目前应用的状况, 并在此基础上提出了以后的研究和应用方向。

关键词: UDT 网络传输协议 长距离高带宽网络 拥塞控制

1 引言

1.1 文章安排

本文第 2 节介绍 UDT 协议的特点。第 3 节给出 UDT 与其它协议的比较和性能分析。第 4 节给出 UDT 协议的不足和改进。第 5 节给出结论以及未来的工作。

1.1.1 基本介绍

随着互联网规模的不断增长, 网络技术的出现, 快速长距离网络、高带宽延迟乘积网络等各种高速网络被广泛使用。目前, 这些高速网络主要用于教育科研等项目, 支持端到端的大数据量的传输, 如在天文、高能物理和生命科学等方面传输实时的数据和图像、媒体文件、语音及各种虚拟环境。高速网络具有广泛的应用前景, 但是现行 TCP 拥塞窗口机制在高带宽长距离的环境下不能很好地工作。首先, 拥塞窗口增加的过于缓慢直接导致吞吐率不高, 无法充分利用带宽。其次, 大的拥塞窗口需要很低的丢包率, 这在现行 TCP 的协议中难以达到。还有就是目前的 TCP 拥塞控制算法在长距离高带宽环境下具有较差的 RTT 公平性。因此, 为了解决网络稳定高效运行的瓶颈, 提出适用于长距离高速网络的传输协议成为当前的研究热点。

国际上的科研工作者对适合于长距离高带宽网络的改进协议进行了深入的研究, 并提出了解决方案,

大体上分为两个方向, 一个是基于 TCP 的改进型协议, 另一个是基于 UDP 的改进型协议。TCP 的改进型协议主要有基于包丢失机制的 HSTCP、STCP、BIC-TCP; 基于端到端队列时延变化的 FAST TCP; 基于多比特信息反馈控制机制的 XCP。UDP 的改进型协议主要有: RBUDP, SABUL, UDT。

在对带宽的利用方面, UDP 协议比 TCP 协议已具有优势, UDP 的不足之处在于其不可靠性, 但是改进型的 UDP 协议既克服了此缺点又保证了传输效率, 因此对改进型 UDP 协议的研究也越来越重视, UDT 正是其中的代表。

2 UDT协议的特点

UDT 建于 UDP 之上, 并引入新的拥塞控制和数据可靠性控制机制。UDT 是面向连接的双向的应用层协议, 支持可靠的数据流传输和部分可靠的数据报传输。更加具体的可以参见文献[1]。

2.1 拥塞控制

UDT 的拥塞控制机制包括速率控制和窗口控制。其中速率控制是主要的, 用于调整发送包的周期, 基于效率、公平和稳定性的综合考虑获得一个发送周期的初始经验值 0.01 秒。当周期内丢包率小于阈值时,

^① 基金项目: 中国科学院知识创新工程青年人才领域项目(O814061101)

收稿时间: 2009-02-18

下一个周期内发送的包将增长

$$inc = \max(10^{\lceil \log_{10}((B-C) \times MSS \times 8) \rceil} \times \beta / MSS, 1 / MSS)$$

当发送方收到一个 NAK 包时, 发送周期将要增长 1/8(即发送速率降低 1/9)。值得特别指出的是, UDT 的速率控制使用的是 DAIMD 算法。窗口控制是辅助性的, 用于动态的限制 NAK 包的数量。速率控制用于快速的达到带宽的最大阈值、快速的丢包恢复及协议内的公平性。窗口控制帮助减少包的丢失和振荡, 避免拥塞。这两种机制共同作用保障了传输的高性能和稳定性, 提高了效率及公平性。

2.2 带宽估计

UDT 使用对包(PP -- Packet pair)的机制来估计带宽值。即每 16 个包为一组, 最后一个是对包, 即发送方不用等到下一个发送周期内再发送。接收方接收到对包后对其到达时间进行记录, 可结合上次记录的值计算出链路的带宽(计算的方法称为中值过滤法), 并在下次 ACK 中进行反馈。

2.3 可靠传输

发送端和接收端都维护一个丢包链表, 用于记录丢包的序号。接收方周期性的发送 ACK 和 NAK 通告发送方丢包信息。从而实现数据传输的可靠性。根据拥塞发生严重时丢包是大量连续的特点, 使用了较好的数据压缩机制, 一个丢失链表中的节点表示一个丢包事件, 其丢包报告中存放的 32-bit 信息如果首位是 0, 则只代表其自身的包丢失, 若是 1 则表明从当前的包序号开始到下一个记录中的所有包都丢失了。例如: 0x00000002, 0x80000006, 0x00000008, 0x0000000B 则表明序号为 2, 6, 7, 8, 11 的包都丢失了。

3 UDT与其它协议的比较和性能分析

3.1 UDT 协议与其它改进型协议的比较

3.1.1 拥塞控制方面

HSTCP、STCP、BIC-TCP^[2-4]在拥塞控制上采用的方法都是改变 TCP 协议的参数尤其是加大窗口的值, 但是每种协议加大或减小参数的方法又有所不同。例如: HSTCP 和 STCP 都是通过改进 TCP Reno 的窗

口变化因子(采用的加法因子和乘法因子不同), 以达到更快的拥塞窗口增加速率和相对较慢的拥塞窗口减小速率; BIC-TCP 使用二进制搜索方法, 以最小的信息损失快速实现最大的网络数据传输能力。

FAST TCP^[5]摒弃了基于目标值的窗口调节方式, 采用排队延迟和报文丢弃一起作为拥塞控制信号, 使得 FAST TCP 在每个 RTT 内报文的增加速率更快, 在收到拥塞信号后更加平滑地减小窗口, 另外排队延迟的特性与网络容量动态特性成正比, 因此更有助于在网络速率增加的情况下提高稳定性。XCP 基于路由路由器的反馈信息而获得整个链路的情况, 此协议提供了一个很好的思路, 但其实现涉及到路由器和终端系统的改变, 操作性不强。

RBUDP^[6]利用 UDP 协议传输数据, 利用 TCP 传输连接和控制信息。它可以非常充分的利用带宽, 避免了 TCP 对每个包进行确认的性能瓶颈, 但是它只能用在专有网络或者支持 QoS 的网络中——因为它没有流控制机制, 并且必须由用户调整发送的速率(在使用前要用 app-perf 对网络带宽进行测量)。

UDT 协议很好的借鉴了以上协议的优点, 如在启动阶段使用 DAIMD 算法, 思路与 HSTCP、STCP、BIC-TCP 相一致, 但窗口增加的方式是递减式的增加, 达到了克服“慢启动”的目的同时又减少了发生拥塞的机率。然而 DAIMD 调整速率的方式是基于时间的, 这点又吸收了 FAST TCP 的优点, 使得在大部分情况下达到最大带宽的时间是一个常量值为 7.5 秒, 提高了稳定性。在对带宽的高效利用和实现可靠传输上, UDT 既实现了 RBUDP 的优点还避免了 RBUDP 的不足。

3.1.2 协议的应用方面

UDT 是应用层的协议, 它在配置和实现方面更加容易, 而其它基于 TCP 的改进型协议都要加载到内核中。很重要的一点是, UDT 已经发展成为一个非常优秀的开源项目, 提供了 C++库和多种工具包, 目前已到第 4 版, 可以在实际中得到运用、发展和检验。UDT 应用十分广泛, 如: 网络电视、文件共享系统、大型分布式环境、网格环境中的数据传输。基于 UDT 的著名开源项目有 Sector and Sphere, PowerFolder,

GridFTP 和 UDT Gateway System 等, 商用项目有 Movie2Me, NiFTy TV, Tideworks [7]。

3.2 性能分析和比较

图 1 是搭建的测试环境

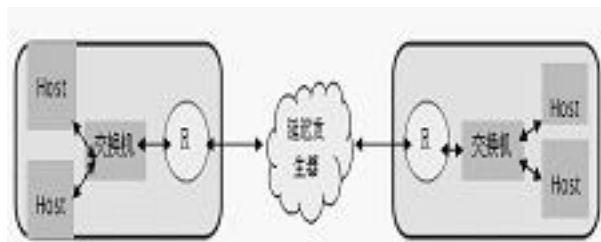


图 1 实验环境

借助于 eVLBI 项目的网络环境, 我们还在北京到上海的 1 G 三层交换链路和北京到香港的 2.5G 光网络中进行了实际的测量。通过对多次测量值的综合, 得到如下结果:

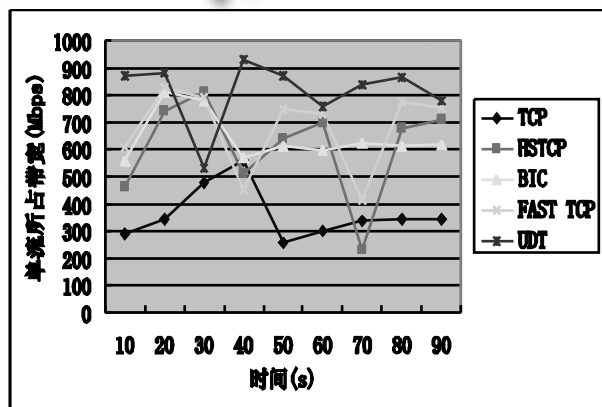


图 2 单流所占带宽曲线

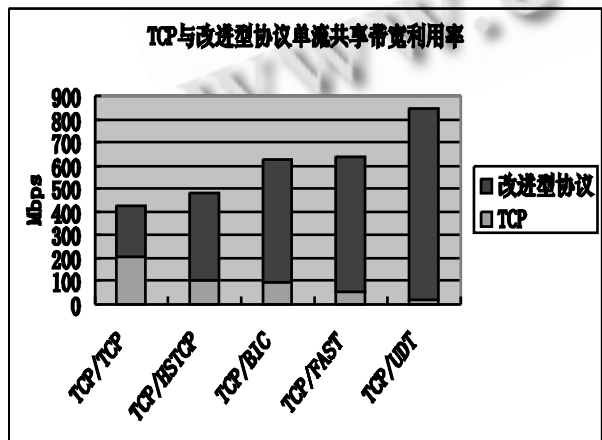


图 3 TCP 与改进型协议单流共享带宽利用率

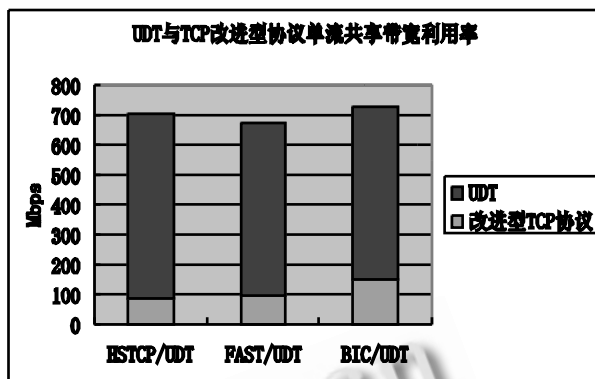


图 4 UDT 与 TCP 改进型单流共享带宽利用率

通过对上面图表的分析, 可以得到如下结论:

基于改进型的 TCP 协议在单流传输时对带宽的利用率只能达到 60% - 70%, 而基于 UDT 协议的单流则可以达到 80% 以上, 因此在对带宽的利用上 UDT 协议有优势。

当链路上有 TCP 流时, 各改进型协议均会出现公平性问题, UDT 对带宽的抢占更为激进。UDT 与其它改进型 TCP 协议流共享链路时, 也会出现协议间的公平性问题。

因此, 如需在一定时间内传输大量数据, 选择 UDT 协议更为合适。

4 UDT协议的不足和改进

UDT 在链路中只有单个流的情况下可获得较高性能, 但是当多流存在时性能有较大的下降。这是由于 UDT 的带宽估计机制在只有单流时对带宽低估, 在多个流时对带宽高估, 低估的情况对性能影响不大, 高估则会对公平性有较大影响并加大了拥塞的机率 and 程度。同时, DAIMD 中带宽的估计对过去的平均值依赖很大, 如果估计带宽有误则不能很好的反映出链路的真实状况。文献[8]提出了 DAIMD 算法和传输队列的改进方法, 但在文中对改进后的如何取值没有详细的阐述。因此, 对拥塞控制算法更加深入的研究和不断的改进始终是协议研究的一个重点。

在应用层面上看, UDT 做服务端接收连接时, 总是新开一个端口与客户端进行连接, 较多客户端连接上来时, 服务端中新打开的众多端口中可能有

的端口会被防火墙拦截而导致通信失败。由于 UDP 端口数最大值有限, 所以 UDT 服务器端可接收的连接数也会受限。还有就是防火墙的穿透, 系统利用率等问题。因此, 仍需对 UDT 开源项目和协议进行不断的改善。

5 总结

本文通过对比 UDT 协议与其它适用于长距离高速网的改进型 UDP 和 TCP 协议, 指出 UDT 协议在大数据量传输上更为合适。同时, 也指出了其在多流共享带宽和目前应用中的不足, 为今后研究和实际应用提供了努力的方向。

参考文献

- 1 Gu YH, Grossman RL. UDT: UDP -based data transfer for high-speed wide area networks. Computer Networks 2007,51:1777 - 1799.
- 2 Floyd S. High speed TCP for large congestion window. RFC 3646, IETF, Dec 2003.
- 3 Kelly T. Scalable TCP: improving performance in high speed wide area networks. ACM SIGCOMM Computer Communication Review, 2003.
- 4 Li YT, Leith D, Shorten RN. Experiment evaluation of TCP protocols for high-speed networks networking. IEEE/ ACM Transactions, 2007,15(5):1109.
- 5 Jin C, Wei Dx, Low SH. FAST TCP: From Theory to Experiments. Network IEEE on Volume 19, Issue 1, Jan-Feb 2005:4 - 11.
- 6 He E, Leigh J, Yu O, DeFanti TA. Reliable Blast UDP Predictable High Performance Bulk Data Transfer. IEEE Cluster Computing 2002, Chicago, Illinois, Sept 1, 2002.
- 7 <http://sourceforge.net/projects/udt>
- 8 Noro M, Tameshige F, Baba K, Shimojo S. Performance Evaluation of Advanced High-Speed Data Transfer Methods in Long-Distance Broadband Networks. 22nd International Conf on Advanced Information Networking and Applications Workshops, 2008:533 - 540.