第37卷 第14期
Vol.37 No.14

计 算 机 工 程
Computer Engineering

2011年7月
July 2011

·网络与通信·

文章编号：1000—3428(2011)14—0094—03 文献标识码：A 中图分类号：TP309

高速长距离网络传输性能优化

王伟杭[1,2]，任勇毛[1]，岳兆娟[1,2]，李 俊[1]

(1. 中国科学院计算机网络信息中心，北京 100190；2. 中国科学院研究生院，北京 100049)

摘 要：从传输协议和网络节点两方面分析高速长距离网络传输性能的影响因素，介绍中间节点拥塞避免、减少主机负载以及改进传输协议等各种性能优化方法，并结合仿真和实际网络实验验证，指出各种技术的优缺点。对传输性能优化技术进行总结并给出设计终端性能自适应的传输协议。

关键词：高速长距离网络；性能优化；传输协议；拥塞控制

Transport Performance Optimization for Fast Long-distance Network

WANG Wei-hang[1,2], REN Yong-mao[1], YUE Zhao-juan[1,2], LI Jun[1]

(1. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China;
2. Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

【Abstract】This paper intensively analyzes the influence factors from the aspects of transport protocol and network node, and emphatically describes all kinds of optimization mechanisms such as congestion avoidance in media node, reducing terminal load and enhancing transfer protocols, etc. Meanwhile, it demonstrates the effectiveness of all these techniques in emulations and real networks and points out their advantages and disadvantages. All these optimization technologies are concluded and a novel research direction designing terminal performance transport protocol is given.

【Key words】Fast Long-distance network(FLDnet); performance optimization; transport protocol; congestion control

DOI: 10.3969/j.issn.1000-3428.2011.14.030

1 概述

近年来，由于DWDM光网络、10 GbE、100 GbE高速以太网等技术的快速发展，骨干网络带宽迅速提升，促使高速长距离网络(Fast Long-distance network, FLDnet)的快速发展。在基于中美俄环球科教网络(GLORIAD)的实际应用和性能测量实验中发现了高速长距离网络传输性能不高的问题，本文对此问题进行了研究，并总结了解决此问题的各种性能优化技术。

2 FLDnet的传输瓶颈

随着高速长距离网络研究的不断深入，研究人员发现，在FLDnet环境中，TCP性能很差[1]。这一问题的产生是由多种因素决定的，总的来说可将原因归结为两方面：(1)传统的TCP协议，采用保守的加性增加和激进的乘性减少的拥塞控制策略，在较大的往返时延(Round Trip Time, RTT)环境中，慢启动和拥塞恢复阶段带宽利用率极低；(2)除TCP协议本身外，网络中间节点及边缘主机的处理能力对高速网络传输也有重要的影响[2-3]。近年来，针对TCP协议的改进以及对节点的性能提高一直是高速链路传输性能优化技术研究的重点。

2.1 传输协议性能瓶颈

针对低速低时延的分组交换网而设计的TCP协议，在FLDnet中性能很差。

(1)拥塞避免机制过于保守。TCP采用的加性增加、乘性减少(Additive Increase Multi- plicative Decrease, AIMD)拥塞窗口调整算法过于保守。在FL- Dnet中，由于RTT较大，一旦发生拥塞，拥塞窗口减小后，需要很长时间才能恢复。文献[1]指出，在带宽为622 Mb/s、RTT为300 ms、报文段大小为1 460 Byte时，TCP拥塞避免阶段所经历的时间长达41 m，如此长的拥塞恢复时间导致TCP传输速率较慢。

(2)流量控制机制过于保守。为避免接收端的缓冲区溢出，TCP使用滑动窗口限制发送流量。默认的最大窗口大小只有64 KB，对于低速低时延网络和早期的低性能终端，这个值较合适，但对于FLDnet，带宽时延积(Bandwidth Delay Production, BDP)远大于这个值，链路管道容量利用率很低。另一方面，目前终端性能已经大大提高，内存早已达到2 GB、4 GB等容量，64 KB大小的接收缓冲区过于保守。

2.2 网络节点性能瓶颈

用于LAN连接的10 GbE物理层规范(LAN PHY)在MAC层可达到10 Gb/s的数据速率，而用于WAN连接的OC-192c/STM-64c光链路(WAN PHY)传输以太网数据的速率，则只有9.286 Gb/s，这由OC-192c的容量及SONET/SDH帧封装开销决定。由于TCP采用基于窗口的发送速率控制机制(而非基于速率的调整策略)，以底层能达到的最大速率发送窗口所允许的数据量，突发数据在LAN PHY与WAN PHY间转换节点由于接入端速率(10 Gb/s)大于广域端速率(9.286 Gb/s)，可能导致缓冲区溢出。因此，FLDnet中LAN PHY与WAN PHY间的节点，成为传输的性能瓶颈[2]。

基金项目：国家科技计划“ITER计划专项”基金资助项目(2008GB111000)；2009年度中国科学院研究生科技创新基金资助项目

作者简介：王伟杭(1985－)，女，硕士研究生，主研方向：高速网络；任勇毛，助理研究员、博士；岳兆娟，博士研究生；李 俊，研究员、博士生导师

收稿日期：2010-12-03 **E-mail：**weihang.wang09@gmail.com

除中间节点外，端到端吞吐率还受限于终端主机产生、发送、接收及处理数据的能力[3]，因此，终端硬件速率(如网络适配器速率、I/O 总线带宽、CPU 处理速率、前端总线速率等)对吞吐率有影响。TCP/IP 协议处理在终端主机进行，对 TCP/IP 协议栈的处理占用大量 CPU 资源。文献[4]指出终端系统处理 TCP 产生的 CPU 开销分为 3 个方面：(1)中断操作开销，包括每个报文达到 NIC 及传输结束等触发的中断操作；(2)数据复制开销，包括在 NIC 存储器与内核空间、内核空间与用户空间之间移动数据等所产生的复制开销；(3)协议处理开销，包括执行 TCP/IP 协议代码及相关的存储管理。在高速网络环境中，这些开销会占用过多的 CPU 时间和存储总线带宽，大大降低终端系统对其他应用的处理能力。

3 网络节点性能优化

3.1 中间节点拥塞的避免

为避免 LAN PHY 与 WAN PHY 间转换节点发生拥塞，应避免短时间内突发数据充满瓶颈节点的缓冲区。

3.1.1 应用层速率抑制

抑制应用程序为 TCP 栈供应数据的速率，是一种对输出到网络的数据速率进行控制的简单方法。这种方法将应用程序提供数据的速率设置为一个低于整个通路瓶颈带宽的恒定值。然而，在 TCP 慢启动阶段，窗口较小，快速到达 TCP 栈的数据在 TCP 传输队列中排队，当有 ACK 到达时，TCP 立刻从队列中弹出数据并发送。排队使得数据进入边缘网络的速率为 10 Gb/s，LAN PHY 与 WAN PHY 间节点仍然是整个通路的瓶颈。

3.1.2 接收窗口大小限制

将窗口值设定为一个较小的值，可抑制 TCP 的传输速率。文献[2]利用一个仿真的 FLDnet，将 TCP 窗口值设为 200 MB。在端到端 10 GbE 环境中，以 1 000 ms 为观测间隔的吞吐率被抑制为约 2 Gb/s，而以 1 ms 观测间隔的吞吐率达到 9 Gb/s。这说明，即使窗口大小被置为一个很小的值，仍无法避免突发数据。大量的传输集中在每次 ACK 到达后，如果不能在整个 RTT 中均衡数据传输，那么就不能避免突发数据引起的报文丢失。

3.1.3 包间间距长度控制

通过设定包间间距(Inter Packet Gaps, IPG)，传输速率被抑制为不超过由下面公式定义的带宽：

suppressed bandwidth=(full bandwidth)×(frame size)/((frame size)+(IPG length))

其中，frame size 指的是以太网帧从前同步信号(preamble)到帧校验序列(Frame Check Sequence, FCS)的长度。利用这一方法，可以有效地控制 LAN PHY-WAN PHY 转换节点处的报文到达速率。

3.2 边缘主机 CPU 负载减少

尽管终端主机可以通过升级 NIC 速率、I/O 总线速率以及前端总线速率等来提高传输性能，主机上的协议处理仍然占用相当多的 CPU 资源。为获得硬件资源的最大性能，终端系统应最小化协议处理产生的 3 种开销。

3.2.1 中断减少

网络速度越快，报文到达越频繁，相应的中断开销就越大。必须寻找有效方法，尽可能减少中断操作发生的次数。

(1)报文段大小增加。每当报文到达 NIC 或传输结束，网络驱动程序都会通知内核去处理，产生中断。对于一定量的数据，采用较大的报文可减少报文个数，从而减少中断次数。

(2)Packet Coalescing。网络接口卡的 Packet Coalescing 功能抑制中断，使内核在一次中断处理过程中从 NIC 获取多个报文。对于发送端主机，从 NIC 中取出的多个报文是 ACK 报文。而对于接收主机，合并的多个报文是数据报文。合并报文数目设置不宜过多，因为过多的数据报堆积在 NIC 的接收缓冲，可能导致缓冲溢出。

(3)Scatter gather I/O。Scatter gather 是与 Block DMA (Direct Memory Access)相对应的一种 DMA 方式。Block DMA 方式传输完一块物理连续的数据后立即发起一次中断，而 scatter gather 方式则不同，它用一个链表描述物理不连续的存储器，然后把链表首地址告诉 DMA Master，DMA Master 传输完一块物理连续的数据后，并不发出中断，而是依据链表继续传输下一块物理连续的数据，直到所有数据传输完毕，最后发起一次中断。显然，scatter gather I/O 方式的中断次数要少于 Block DMA 方式。

(4)Delayed ACK。Delayed ACK 是一种减少 ACK 个数的方法：接收端主机发送一个 ACK 确认多个数据报的到达。目前，在 Linux 的默认设置中，每个 ACK 最多允许确认 2 个数据报，因为过分延迟 ACK 会影响及时的拥塞控制。

3.2.2 复制减少

在数据传输过程中，发送端执行发送需经历 4 个步骤：(1)应用将数据放入用户空间；(2)CPU 将数据从用户空间复制到内核空间；(3)CPU 加载全部待发送的数据，计算检验和；(4)数据从内核被复制到 NIC，如果 NIC 不提供 DMA 方式，复制由 CPU 来操作。

与发送端对应，接收端也需要类似的数据复制过程。简单而大量的复制操作占用了宝贵的 CPU 时间，因此，如何减少复制次数，实现零复制(zero copy)，一直是研究人员关注的热点。

(1)DMA(Direct Memory Access)。DMA 方式的基本思想是计算机内的硬件子系统，不需要或很少需要 CPU 的干预，直接与内核空间打交道，DMA 控制器代替 CPU 来控制内核与 I/O 之间的数据交换。可以使用 DMA 方式的硬件系统包括声卡、显卡、网卡等。

(2)RDMA(Remote Direct Memory Access)。RDMA 协议在发送端将目标地址信息插入到报文中，接收端 NIC 直接将数据放置到地址信息所指定的用户位置，避免了数据在用户空间与内核之间的多次复制。

(3)检验卸载。在发送、接收主机执行检验和计算时，需要 CPU 加载全部数据并进行一系列加运算。为减少这项操作的 CPU 开销，可在 NIC 与内核交互数据时，由 NIC 硬件执行检验和计算。

(4)复制/检验合并。在报文的一次移动中合并多项操作可以减少复制次数。例如，一些 TCP/IP 实现在数据复制到(或离开)用户空间时，同时执行检验和的计算。

3.2.3 软件执行处理减少

主机处理 TCP 占用了大量的计算资源，导致用于其他应用的处理资源大大减少。用专门的硬件代替 CPU 实现协议处理，可以降低 CPU 占用率。

(1)TSO(TCP Segmentation Offload)与 GSO(Generic Segmentation Offload)。TSO 和 GSO 是用硬件代替 CPU 将数据分为一个个数据报，为它们产生报头，并计算校验和。硬件计算校验和要比软件执行更有效。

(2)LSO(Large Send Offload)。如果一个 TCP 数据报大于整条通路所允许的最大传输单元(MTU)，则 LSO 使用硬件将

这个 TCP 报文分解为多个小的数据报。

(3)LRO(Large Receive Offload)。与 LSO 相对应，在接收的数据报到达内核之前，LRO 利用硬件把多个小的 TCP 报文合并为一个较大的报文。

(4)TOE(TCP Offload Engine)。TCP 协议卸载引擎技术(TOE)的基本思想是将 TCP/IP 协议的全部处理或部分处理转移到专门的硬件上，从而减轻主机 CPU 的负载。同时，在专门的硬件处理协议时，不必将用户数据复制到内核空间，减少了数据复制开销。

4 传输协议性能优化分析与实验验证

由于 TCP 协议自身的缺陷，近年来，研究人员提出了各种各样的改进方案。本文将改进方案分为 3 类：(1)沿用现有的 TCP 协议；(2)采用各种改进的高速传输协议；(3)采用 TCP 加速器产品。

4.1 现有 TCP 协议沿用

这类方法技术实现难度最小，通过修改 TCP 协议的默认参数或采用多个 TCP 流传输来提高带宽利用率。

4.1.1 TCP 窗口值增大

标准 TCP 窗口大小为 64 KB，对于 RTT 为 300 ms 的高延迟网络，单个 TCP 连接吞吐率最大只有：

$$Throughput = W/RTT = 64\ \text{KB}/300\ \text{ms} \approx 1.7\ \text{Mb/s}$$

由于 TCP 窗口大小的限制，在高延迟网络中，传输速率存在瓶颈，因此对 TCP 窗口大小的默认值进行修改，可以显著提高 TCP 流传输速率。TCP 最大传输速率受限于如下公式：

$$Throughput \leqslant \frac{W}{RTT}$$

其中，W 为窗口大小；RTT 为往返时延。为使吞吐率尽可能接近带宽速率，应调整窗口大小，使其满足：$W \geqslant BDP$，BDP 为带宽时延积($Bandwidth \times RTT$)。本文实验测量了不同窗口大小情况下单个 TCP 连接的吞吐率，如图 1 所示。

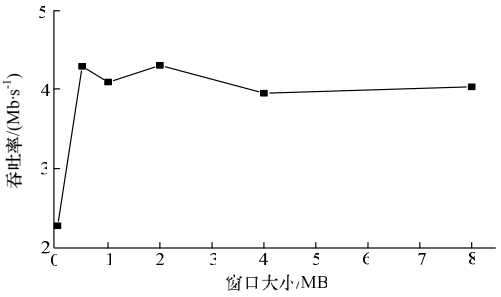

图 1 TCP 窗口大小对吞吐率的影响

实验发现，更大的窗口值能提高吞吐率，但是，到达一定值后，窗口值再增大反而导致吞吐率下降，本文分析这是由于终端系统的处理能力有限所致。

RFC 1323 对 TCP 协议进行改进，增加了窗口扩大选项(Window Scaling Option)，使最大窗口大小扩大到 $2^{30} \approx 1$ GB，如果使用此最大窗口值，则吞吐率最大值约为 26.7 Gb/s。

4.1.2 最大传输单元增大

报文段大小对于吞吐率有影响。TCP 采用最大报文段大小(Maximum Segment Size, MSS)选项来允许接收端通告它愿意接收的最大报文段大小。如果报文段太小，那么 TCP/IP 头开销占用比例过大，带宽利用率低，因此，采用大的 MSS 传输可提高吞吐率。如图 2 所示的实验发现，对于单个 TCP 连接，采用 8 960 Byte 的 MSS 与 536 Byte 相比，最高吞吐率提高约 4 倍。但是，目前一些网络设备并不支持巨型帧。如果报文段过大，超过链路层的最大传输单元，IP 包就需要进行分片，只要其中任一分片丢失，就必须重传此报文段的所有分片。因此，分片使得报文丢失的概率变大，重传所有分片导致性能下降。另外，这种方法需要沿途所有路由器和交换设备的支持，很难得到大规模的实际部署。

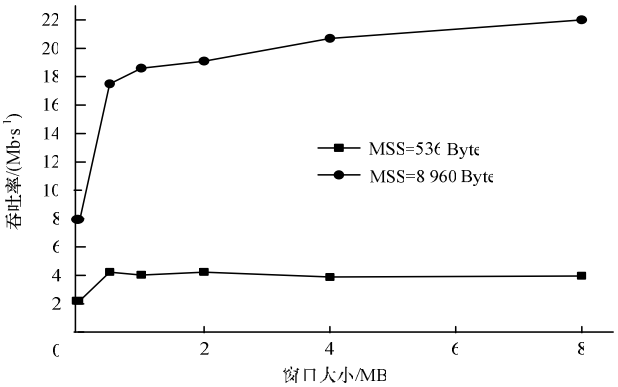

图 2 采用不同 MTU 时的吞吐率

4.1.3 多个并行流传输

多个 TCP 流并行传输能大幅提高传输速率。实验发现，对于相同 buffer 大小的多个流传输，吞吐率随着并行流数增多而提高，在不改变其他参数的条件下，采用 100 个并行流能提高吞吐率约 71 倍。但是，由于终端主机处理性能限制，并行流不能同时运行太多，因为 TCP 需同时维持过多的会话连接，增加了 TCP 处理的复杂性，导致处理效率下降。

4.2 改进的高速传输协议

在众多的高速协议改进方案中，研究得最多的是对 TCP 协议进行改进。最近几年，基于 UDP 的高速传输协议也不断涌现。

4.2.1 高速 TCP 改进协议

高速 TCP 改进协议主要针对 TCP 的拥塞控制机制进行改进。在这些改进方案中，大量的研究集中在对 AIMD 窗口算法的调整，有代表性的改进协议包括：HSTCP(High Speed TCP)、BIC TCP(Binary Increase Congestion TCP)、CUBIC 等。另外一些著名的改进算法，如 FAST、XCP 和 VCP 等对拥塞检测和通告机制进行修改，TCP 协议主要依靠重传定时器超时和重复的 ACK 进行拥塞检测和通告，这种方式在高速长距离网络中不够准确。

各种 TCP 改进协议旨在解决 TCP 协议的低效问题，同时对 TCP 友好性、RTT 公平性以及收敛速度也开展了深入的研究。目前，已经有很多针对现有 TCP 改进协议的分析和性能评价方面的研究。文献[5]采用 NS2 模拟实验方法，通过改变瓶颈带宽、往返时延及 TCP 并行流数等参数，模拟实验了各种 TCP 改进协议的传输性能，在吞吐率、时延和丢包等性能指标上对各种协议进行比较。然而，现有的大多数 TCP 改进协议只局限于对 AIMD 调整算法的一个简单修正，并没有从整个拥塞控制，包括拥塞检测、拥塞通告和拥塞反应的整个过程来考虑。

4.2.2 基于 UDP 的高速传输协议

由于 TCP 的窗口机制在 FLDnet 中效率很低，研究人员提出了基于 UDP 协议的改进方案。这类改进协议的核心思想在于将数据和控制信息分开传输，利用 UDP 传递数据，使用 TCP 交换控制信息。在无连接的 UDP 之上提供了可靠的数据流服务，同时消除了 TCP 基于窗口的流量控制。有代表性的协议包括：RBUDP，UDT 和 Tsunami 等。实验比较 RBUDP、

(下转第 99 页)