计算机科学 2008Vol.35№.8

基于 FPGA 的 TOE 系统设计与实现*)

王 圣 苏金树
(国防科大计算机学院 长沙 410073)

摘 要 随着网络带宽的迅速增长,主机协议处理开销已经成为系统整体性能的瓶颈。为了有效增加系统吞吐率,进一步减轻 CPU 的负担,本文详细描述了一种基于 FPGA(Field Programmable Gate Array)的 TOE(TCP Offload Engine)系统的设计与实现。实验结果表明,系统在吞吐率等方面明显优于非 TOE 系统。

关键词 FPGA,TOE,设计

Design and Implementation of FPGA-based TOE System

WANG Sheng SU Jin-shu
(Computer School,NUDT,Changsha 410073,China)

Abstract With the rapid increment of network bandwidth, the overheads of protocol on host have become the bottleneck of system performance. To increase the throughput efficiently and alleviate the burden of CPU, we design and implement a system based on FPGA (Field Programmable Gate Array) in detail, called TOE (TCP Offload Engine), which places the main part of TCP into hardware. Results show that TOE outperforms non-TOE in terms of throughput to TCP flows.

Keywords FPGA,TOE,Design

1 简介

随着网络技术,尤其是光纤技术的快速发展,光纤通信网络正迅速成为主要的网络传输手段,使网络带宽不断提升。网络应用的性能需求表现为高吞吐量、低延迟、高带宽、低主机开销和低存储开销等特点。当 100Mbps 以上网络出现后,网络协议处理一直占用相当多的 CPU 能力。对于万兆以太网[1]或 Infiniband[2],主机系统上的协议处理更是成为网络可靠传输的性能瓶颈,所以 TCP(Transmission Control Protocol)加速技术一直是近年来的研究热点。经过研究,人们发现 TCP 处理的主要开销分为中断操作、数据复制和协议处理这 3 个方面[3]。针对这些问题,许多专家研究如何提高 TCP 性能,如优化校验和计算、采用高速缓存实现 PCB(Protocol Control Block)、进行首部预测等[4];设计新型的拥塞控制机制,尽量减少或者避免拥塞[5];改变 TCP 实现的结构,将部分或全部功能卸载到硬件,即 TOE[6]。由于硬件具有执行速度快的优点,因此 TOE 能够获得大幅度的性能提升。在具体实现 TOE 时,可以使用可编程器件或通用处理器,以充分获取灵活性,或者使用 ASIC 设计,以获取更高的性能。关于 TOE 的实验结果表明,主机 CPU 的占用率在传输大容量报文时明显有所降低[7]。

本文的组织如下:第 2 节阐述 TOE 系统的设计思想,介绍基于 FPGA 的 TOE 系统的实现,并给出相应的实现框架;第 3 节给出系统的性能;最后总结全文。

2 TOE 系统设计与实现

2.1 TCP 性能瓶颈

设网络带宽为 $B_{network}$,则单位时间内报文数目为 $y=\frac{B_{network}}{L_{packet}}$,其中,$L_{packet}$ 为报文平均长度。设 O_{cpu} 为 CPU 处理网络流量开销,此开销将包含协议处理 O_p 与中断处理两部分 O_{int}:

$$O_{cpu}=O_p+O_{\text{int}} \tag{1}$$

其中,协议开销与且只与报文数目相关:

$$O_p=\phi_p\cdot y=\phi_p\cdot\frac{B_{network}}{L_{packet}} \tag{2}$$

其中,ϕ_p 为单个报文的协议处理。中断开销则包含数据传输与中断处理:

$$O_{\text{int}}=\left(\alpha_d+\frac{\beta_{\text{int}}}{\eta_{\text{int}}}\right)\frac{B_{network}}{L_{packet}} \tag{3}$$

其中,α_d 为数据传输开销。如果采用 DMA 传输,则 α_d 为相应的启动开销,而 β_{int} 为硬中断开销,η_{int} 则为中断合并常量,称为中断阈值,即合并若干报文后才发出接收中断,以减少中断开销。

综合(1)、(2)、(3)式,得 CPU 开销与能够处理带宽之间的比值,即费效比 γ:

$$\gamma=\frac{O_{cpu}}{B_{network}}=\frac{1}{L_{packet}}\left(\phi_p+\alpha_d+\frac{\beta_{\text{int}}}{\eta_{\text{int}}}\right) \tag{4}$$

从(4)式可以看出,为了使 γ 尽可能小,可以增大报文长度;减小协议处理时间;使用 DMA 机制减少传输开销;实现中断合并策略减少中断开销。硬件实现 TCP 将能在这些方面提供显著的改善。

需要说明的是,公式(4)并未提及存储带宽。实际上,存储带宽需要 N 倍于网络带宽,其中 N 为复制关联常量,与系统所采取的复制策略有关。如果按照普通的处理协议,则存储带宽需为网络带宽的 6 倍[8]。即便采用 0 复制技术,网络

*)基金项目:获国家自然科学基金资助,项目编号为 90604006。王 圣 博士研究生,主要研究方向为计算机网络技术与信息安全;苏金树 教授,博士生导师。

数据最少也需经过 2 次存储总线，因此 $N \geqslant 2$。

2.2 TOE 系统设计

为了有效减少协议处理时间，减轻 CPU 的开销，同时为了减小硬件设计难度，系统采用了软件负责连接的建立与撤销，硬件负责数据传输的机制，对复杂度与处理卸载进行了有效的平衡，系统框架如图 1 所示。

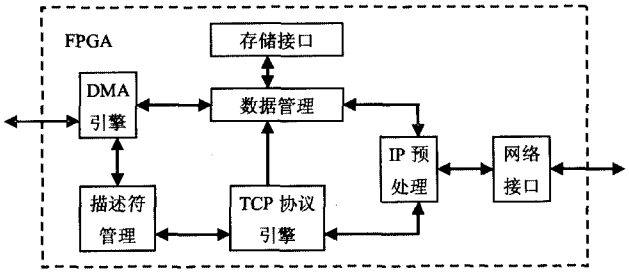

图 1 系统设计框架

为了能够有效处理 TCP 数据的接收，系统采用将 IP 预处理分为接收和发送两部分。接收部分将从 10G 网络接口接收原始报文，并对报文进行初步解析：对报文进行校验和验证，对报文各个部分的长度控制信息进行初步处理。判断报文的一些基本字段，根据报文的控制数据生成相关控制信息，利用这些信息在连接卸载 Hash 表里进行匹配查找。如果匹配成功，则将获得连接控制块的索引，并将控制信息以及报文传递到 TCP 协议引擎进行处理；如果匹配失败，则将原始报文直接传至主机。

发送部分将从 TCP 协议引擎接收数据以及相应的控制信息，构造完整的报文发送至网络接口，或者直接从主机获取非卸载 TCP 报文发送至网络接口。

为了充分有效地利用硬件的并行优势，TCP 协议引擎采用分布式控制策略，通过 TCP 控制变量的解耦分析，将 TCP 连接的主要控制参数分解成接收与发送两部分，如图 2 所示。

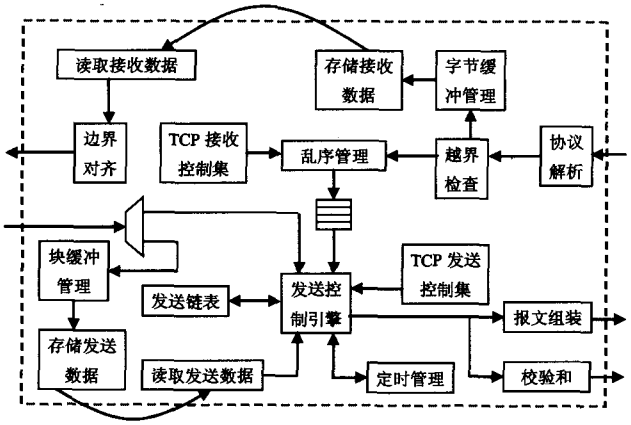

图 2 TCP 协议引擎

TCP 协议处理引擎也分为接收与发送两部分。接收部分根据控制信息得到连接控制块索引，依据此索引在控制缓冲池中查找相应连接的接收控制变量，并从报文中提取 TCP 字段信息，进行乱序处理。根据得到的可靠保序数据生成相应的控制信息，最终有效数据经过存储接口进行边界对齐后传递给主机。发送部分分为三个处理要素。首先从主机接收用户数据，依次以块为单位存储在发送缓冲区中，构成发送队列，在适当条件下根据控制信息生成相应的报文；其次从 TCP 接收信息中获取对端的控制信息，例如获取应答序列号，更新本地的发送队列，根据窗口字段计算对端的接收窗口大小，并从 TCP 发送信息中获取本地的可靠保序序列号，生成相应的应答序列号，以及本地的剩余窗口空间构造发送报文的窗口字段等；最后，响应定时器消息，进行报文的重传与探测控制。

为了充分发挥硬件的并行性，系统的各个功能部分采用 FIFO 缓冲相连接，以降低相互之间的耦合度。

2.3 FPGA 实现

系统采用 Altera 的 EP2S90 实现，EP2S90 共有 72768 个 ALUT(Adaptive Look-u PTable，自适应查找表)，TOE 系统共使用了 54994 个 ALUT，占据芯片总数的 76%。芯片共有存储单元约为 4520448 Bit，TOE 系统使用了 2170488 Bit，大都用作关键数据的存储以及各种部件之间的耦合。

3 性能分析

3.1 实验环境

系统实验采用双服务器，服务器均配置双至强 3.2G，拥有 2M 缓存的 CPU，拥有 4GB 内存，250G 的 SATA 硬盘，操作系统为 Linux，内核为 2.6.1710G，每台服务器上均配置了 TOE 网卡。两台服务器之间通过 Cisco 的 10G 交换机连接。

测试软件采用 IPERF，每次测试时间均为 60s，测试次数为 5 次，取平均值。

3.2 实验结果

当 TCP 缓冲设置为 64k 字节，接收中断阈值设为 63 个，发送中断阈值设为 31 个，即接收到主机 31 个发送请求后才发出发送中断，以便主机系统进行描述符回收。描述符环的长度设为 128。

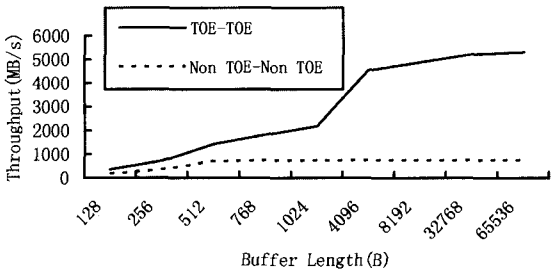

图 3 TOE 显著提升吞吐率

从图 3 还可看出，当应用的发送缓冲设置较大时，使 TOE 的有效报文长度，即 L_{packet} 增加，性能随之得到显著提升。同时，根据(4)式，增大 η_{int} 值，即实现中断合并策略，将会使系统的性能得到改善，如图 4 所示。

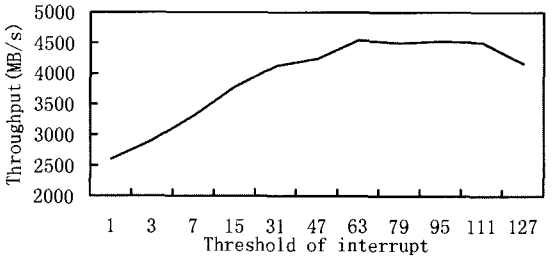

图 4 中断合并改善 TOE 性能

结束语 随着网络带宽的迅速增长，主机协议处理开销已经成为系统整体性能的瓶颈。为了有效增加系统吞吐率，进一步减轻 CPU 的负担，本文详细描述了一种基于 FPGA 的 TOE 系统的设计与实现。实验结果表明，系统在吞吐率等方面明显优于非 TOE 系统。为了进一步减小存储压力，我们将在后续研究中对数据的操作进行优化，并基于此系统进一步研究 TCP 硬件加速技术。

参 考 文 献

[1] Recio R J. Server I/O networks past, present, and future// Proc. of the ACM SIGCOMM Workshop on Network-I/O Convergence: Experience, Lessons, Implications. New York: ACM Press, 2003: 163-178

[2] Intel in Communications. 10 gigabit Ethernet technology overview. White Paper, 2003. http://www.intel.com/network/connectivity/resources/doc_library/white_papers/pro10gbe_lr_sa_wp.pdf

[3] Chase J S, Gallatin A J, Yocum K G. End-System optimizations for high-speed TCP. IEEE Communications Magazine, 2001, 39(4): 68-74

[4] Wright G R, Stevens W R. TCP/IP Illustrated, Volume 2: The Implementation. Addison Wesley, 1995

[5] 章淼, 吴建平, 林闯. 互联网端到端拥塞控制研究综述[J]. 软件学报, 2002, 13(3): 354-363

[6] Yeh E, Chao H, Mannem V, et al. Introduction to TCP/IP offload engine. 2002. http://www.10gea.org/ SP0502 IntroTo TOE_F.pdf

[7] Adaptec Corporation. Advantages of a TCP/IP offload ASIC. 2004. http://graphics.adaptec.com/pdfs/ tcpio_adv_wp.pdf

[8] Minturn D, Regnier G, Krueger J, et al. Addressing TCP/IP processing challenges using the IA and IXP processors. Intel Technology Journal, 2003, 7(4): 39-50

(上接第 47 页)

$\{P_i | i=1,2,3,\cdots,n\}$, $COV(P_i, T)$表示 P_i 投递的探测报文到可以到达的进入 T 的入口点的集合。

探测源选取规则 1 如果 h 分别选择了探测源 $T(h)1$ 进行探测，满足 $E(T(h)1) \notin COV(P,T)$，则将 $T(h)1$ 添加到 P 中。

说明：$E(T(h)1) \notin COV(P,T)$，显然通过 $T(h)1$ 可以到达目标网络新的入口点，选择 $T(h)1$ 加入 P 可以提高拓扑发现的覆盖率。当 $E(T(h)1) \in COV(P,T)$时，则按照规则 2 和 3 继续选择。

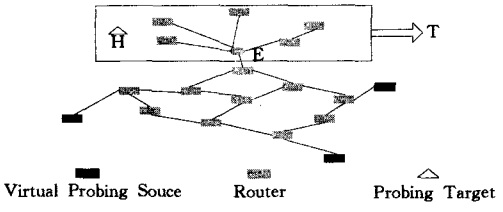

图 2 分布式拓扑探测构想图

探测源选取规则 2 如果 h 分别选择了探测跳板 $T(h)1$ 和 $T(h)2$ 进行路径探测，满足 $E(T(h)1) = E(T(h)2) \notin COV(P,T)$，且 $Hop(h, E(T(h)1)) < Hop(h, T(h)2)$则将 $T(h)1$ 添加到 P 中；

说明：当探测源选择不同的虚拟探测源进行路径探测都到达目标网络的同一个入口点时，则将其中路径最短的情况下选择的探测源加入拓扑发现的探测源集合。

探测源选取规则 3 如果 $Hop(h, E(T(h)1)) = Hop(h, T(h)2)$ 且 $Hopv4(h, T(h)1) < Hopv4(h, T(h)2)$，则将 $T(h)1$ 添加到 P 中。

说明：总路径跳数一样的情况下优先选择 IPv4 路径短的虚拟探测源添加到虚拟探测集合主要考虑到 IPv6 路由表为分级结构，而 IPv4 是扁平结构，所以在跳数相同的前提下报文在 IPv6 网络中的转发更为迅速。

4 实验及结论

为了验证 VDPM 相对单源 IPv6 拓扑探测在提高拓扑覆盖率方面的优势，我们进行了实际拓扑探测，整个过程分为两个步骤：

步骤 1 在 202.196.63.0 子网内设置了两个探测源，分别与上海交通大学的 ISATAP 路由器(2001:da8:8000:3:0:5efe:202.112.26.254)和清华大学的 ISATAP 路由器(2001:da8:900e:0:5efe:59.66.4.50)建立了 IPv6-in-IPv4 隧道；选出 Cernet-2 上分布相对均匀的 120 个可达性探测目标点作为探测目标点集合，进行拓扑探测，在未进行别名归并的前提下，发现路由器接口数为 373，发现链路数为 625。

步骤 2 从河南 Cernet-2 接入中心，前缀为 2001:da8:a5:e::/64 的链路内，设置纯 IPv6 探测源对相同的 120 个探测目标点进行拓扑探测，在未进行别名归并的前提下，发现路由器接口数为 209，发现链路数为 386。

通过对比看出 VDPM 相对单源 IPv6 拓扑探测在提高拓扑覆盖率方面有显著优势，唯一缺点是报文的应答时间过长，平均应答时间都在 500ms 以上。通过对测试数据的进一步分析，我们发现步骤 2 中只有 35.3%路由器接口地址与步骤 1 得到的相关数据重合，因此下一步考虑将纯 IPv6 接入环境下的单源探测方式补充为 VDPM 的一部份，这样可以进一步提高拓扑发现的覆盖率。在后续的工作中，我们将着重分布式探测冗余避免策略的完善，并依据 VDPM 设计出面向国家级的大规模 IPv6 网络拓扑发现原型系统。

参 考 文 献

[1] Astic I, Festor O. A hierarchical topology discovery service for IPv6 networks// IEEE/IFIP Network Operations and Management Symposium (NOMS). Apr. 2002: 497-510

[2] 李云琪, 杨家海. 一个面向 IPv6 的网络拓扑管理系统的实现[J]. 计算机工程与应用, 2004, 40(29): 73-75, 181

[3] 沈曾伟, 周刚. IPv6 接入网拓扑结构自动发现方法研究[J]. 计算机工程, 2006, 19: 136-138

[4] Waddington D G, Chang Fangzhe, Ramesh V, et al. Topology Discoveryfor Public IPv6 Networks[J]. ACM SIGCOMM Computer Communications Review, 2003, 33(3): 59-68

[5] IPv6 Scamper. http://www.wand.net.nz/~mjl12/ipv6-scamper/

[6] Mao Z Q, Rexford J, Wang J. Towards an Accurate AS-Level TracerouteTool // Proc. ACM SIGCOMM 2003. Karlsruhe, Germany, New York: ACM Press, Aug. 2003: 365-378

[7] Vukadinovic D, Huang P, Erlebach T. On the Spectrum and Structureof Internet Topology Graphs// H. Unger, T. Böhme, A. Mikler, eds. Lecture Notes in Computer Science (LNCS) 2346. Berlin: Springer-Verlag, 2002: 83-95

[8] Spring N, Mahajan R, Wetherall D. Measuring ISP Topologies w-ithRocketfuel. ACM SIGCOMM CCR, 2002, 32(4): 133-145

[9] Donnet B, Raoult P, Friedman T, et al. Deployment of an Algorithm for Large-Scale Topology Discovery. IEEE Journal on Selected Areas in Communications (JSAC), Special Issue on Internet, Special Issue on Internet Sampling, 2006, 24(12): 2210-2220

[10] Pansiot J J, Grad D. On Routes and Multicast Trees in the Internet. ACM SIGCOMM CCR, 1998, 28(1): 41-50

[11] Govindan R, Tangmunarunkit H. Heuristics for Internet Ma P-Discovery// Proc. IEEE INFOCOM 2000. Tel-Aviv, Israel, Mar. 2000, 3: 1371-1380

[12] Huffaker B, Plummer D, Moore D, et al. Topology Discovery by Active Probing// Proc. 2002 IEEE Symposium on Applications and the Internet Workshops (SAINT'02w). Nara, Japan, 28 Jan.-1 Feb. 2002: 90-96

[13] Barford P, Bestavros A, Byers J, et al. On the Marginal Utility of N-etworkTopology Measurement// Proc. 1st ACM SIGCOMM Workshop on Internet Measurement (IMW 2001). San Francisco, California, Nov. 2001: 5-17

[14] Meyer D. Route Views Project. University of Oregon Available URL: http://www.routeviews.org/

[15] Spring N, Mahajan R, Wetherall D. Measuring IS PTopologies with Rocketfuel. ACM SIGCOMM CCR, 2002, 32(4): 133-145

[16] Mao Z Q, Rexford J, Wang J. Towards an Accurate AS-Level Traceroute Tool// Proc. ACM SIGCOMM 2003. Karlsruhe, Germany, New York: ACM Press, Aug. 2003: 365-378