

OBS 中 TCP 业务的分段指示拥塞控制策略

田张志, 王文国, 吴国栋, 晁瑞兰, 刘春艳

(曲阜师范大学 计算机科学学院, 山东 日照 276826)

摘要:传统的 TCP 拥塞控制算法主要是针对电通信网络中包交换机制提出的, 当这种拥塞控制算法应用到光突发交换 (OBS) 中会出现很多问题, 甚至会使网络性能急剧恶化。为了改善传统的 TCP 拥塞控制算法在 OBS 网络中的性能, 提出了一种分段指示拥塞控制技术, 它能根据光网络链路的占用情况, 在 OBS 边缘节点随机地标记不同 TCP 流的数据包以阻止网络拥塞。该方法不仅能对网络拥塞迅速地做出反应, 而且还能应对 OBS 的假超时现象 (FTO), 进一步改善 OBS 网络的性能。

关键词:光突发交换; TCP; 拥塞控制; Swarm 仿真

中图分类号: TN915.04

文献标识码: A

文章编号: 1673-629X(2008)09-0076-03

A Stage - Based Congestion Notice Strategy on TCP over OBS

TIAN Zhang-zhi, WANG Wen-guo, WU Guo-dong, CHAO Rui-lan, LIU Chun-yan

(Computer Science College of Qufu Normal University, Rizhao 276826, China)

Abstract: Traditional TCP congestion control algorithm mainly aimed at packets based communication network, when it is implemented in OBS network there will be series of new problems and may even degrade the performance of the OBS network. A three-stage-indication method is first proposed to improve congestion situations of TCP traffic in an OBS network. The technology can randomly mark different TCP flows in OBS border router in order to avoid congestion. It can not only respond quickly when congestion occurs, but also detect the false time-out phenomena, and therefore lead to better performance of the OBS network.

Key words: OBS; TCP; congestion-control; Swarm simulation

0 引言

QIAO^[1]等人提出的光突发交换 (OBS, Optical Burst Switching) 方案已经成为下一代全光互连网的理想交换模式之一, 其网络拓扑图如 1 所示。

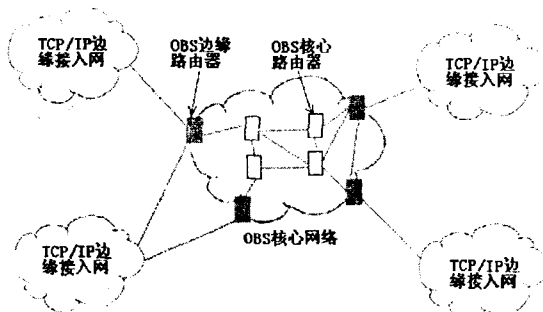


图 1 OBS 网络拓扑图

其原理是, OBS 边缘路由器收集接入网络的 IP 数据包, 把目的地址相同的数据包汇聚 (assembling) 成 Burst, 待组装完成后, 边缘路由器首先发出一个头部控制信息 (BHP, Burst Head Packet) 在核心路由器上预留资源, 经过一段时间 (JET, Just Enough Time) 边缘路由器发出 Burst, 核心路由器根据刚才 BHP 预留的资源透明地传输 Burst。然而一旦在核心路由器两个 Burst 由于竞争资源失败, 必须将其中之一丢弃。由上述 Burst 组装方法, 一个 Burst 往往包含多个通信源的数据包, Burst 的丢失会对所有这些源端造成影响, 特别对于 TCP 业务将会大大降低网络性能。当前网络流量的 80% ~ 90% 属于传输控制协议 (TCP) 所承载的业务, 且这种主宰地位在相当长的时间内不会改变。但是传统的 TCP 控制算法是针对包交换网络而制定的, 包丢失意味着缓冲区溢出; 由于在 OBS 中一般没有缓存且资源是单向预定的, 数据包丢失可能是由于网络阻塞, 也可能是多个突发包竞争信道失败所致。所以当把传统的 TCP 控制算法应用到 OBS 中时就会有新的问题出现: 如果仍按传统网络那样, TCP 发送端由于网络阻塞的原因进入慢启动 (Slow-Start), 则

收稿日期: 2007-12-11

基金项目: 国家人事部高层次留学人员回国工作资助项目 (国人部发 [2004] 61 号)

作者简介: 田张志 (1982-), 男, 山东夏津人, 硕士研究生, 研究方向为光通信网络, 网络行为; 王文国, 教授, 研究方向为光网络通信及网络安全。

势必会降低网络吞吐量。特别是当一个发送轮回(RTT)的所有数据均在一个 Burst 中时将会全部丢失。因为一个 Burst 中含有多个 TCP 发送端的数据,会有多个 TCP 发送端同时进入慢启动,这样网络就会出现全局同步现象,其吞吐量会急剧下滑,网络性能会显著恶化。

文献[2]和[3]的作者通过仿真比较了当前三种流行的 TCP 版本,即 Reno, New-Reno 以及 SACK 在传统网络和 OBS 网络中的性能,指出在快速的 OBS 中 SACK 的性能最好而 New-Reno 的性能最差,并且提出了 BTCP(Burst TCP)的想法。文献[4]对 HSTCP(High Speed TCP)进行了仿真,HSTCP 利用了一种称为 SAIMD(Statistical Additive Increase Multiplicative Decrease)的拥塞窗口控制机制。以上文献一个共同的特点是只考虑了在 OBS 网络中出现拥塞的情况,并没有考虑处于 OBS 边缘的接入网拥塞情况,并且无一例外的将所有的传输控制由 TCP 发送端负责。文中针对以上情况首次提出了一种基于 RED/ECN(Random Early Detection / Explicit Contention Notification)的分段指示拥塞控制机制,将部分控制权移交给 OBS 边缘节点,从而减少了对 TCP 发送端的影响,有助于提高网络的吞吐量。文中提出的算法不仅能有效区别数据丢失的位置,还能探测到假超时(FTO)现象,而且对于接入网络中的数据丢失能做出迅速响应,使网络尽快恢复到最佳状态。

1 分段指示拥塞控制算法

可以想象共有三个地方可能丢失数据:两端的接入网和 OBS 网络。在接入网络中丢失数据与在传统网络中一样是由于缓冲区溢出,表明网络严重拥塞;而在 OBS 中可能是由于拥塞也有可能是由于偶然的冲突(如前所述)。采用分段指示的策略,构造主动拥塞控制算法如下:

(1)在第一段中,TCP 发送段发出数据后,数据首先通过本地接入网络到达 OBS 边缘节点 Ingress。在数据包经过本地网络路由器时,路由器完全根据传统算法^[4]对数据包进行标记。当数据包到达 OBS 边缘节点 Ingress 时,它首先检查数据包的 ECN 位,如果有数据包的 ECN 位为 1,Ingress 则根据数据包的源地址向 TCP 发送端发送抑制消息。这样发送端能更有效地探知本地接入网络的拥塞情况,在本地网络发生拥塞后发送端能更快地做出响应,同时也能防止数据包在 OBS 中特别是对方接入网络中丢失而造成的 ECN 置位失效。

(2)当在 OBS 的 Ingress 组装 Burst 完成时,其根据

Burst 的情况发送 BHP,当 BHP 在 OBS 某一核心节点预留资源失败时,OBS 核心节点判断失败的原因:是由于偶然的 Burst 竞争信道还是由于信道占用率过高,判断的标准可以设定两个阈值 Max_{thresh} 和 Min_{thresh} 。当前信道的占用率小于 Min_{thresh} 时,表示 Burst 的丢失只是由于偶然的抢占冲突,并不能说明网络拥塞,简单地向发送端发送控制信令:重传未发送成功的数据,并且将 TCP 重传计时器清零即可。当信道占用率大于 Max_{thresh} 时,表示网络出现了严重的拥塞,此时整个 Burst 应当丢弃,令 Burst 中所有 TCP 源端重新进入慢启动阶段,并重传前一个发送窗口内的所有数据。当信道的利用率位于两个阈值之间时,可以采用如图 2 所示的概率 P 从 Burst 的 TCP 源地址集中选择,使选中的 TCP 源进入慢启动阶段,未选中的 TCP 源重传前一个发送窗口内未发送成功的数据包,TCP 重传计时器清零,发送窗口不变即并不进入慢启动阶段。

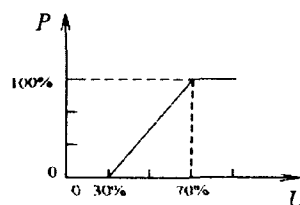


图2 选择概率 P 与信道占用率 U 的关系曲线

除了上述所讲的根据信道的占用率来决定选择多少 TCP 源(即决定 P)需要显示指示的方法外,还可以利用一种即时统计的方法来计算 P ,表述如下:初始平均标识概率 $P = P_0$,初始相邻两个丢失的 Burst 的平均时间间隔为 $T = T_0$ 。当某 Burst 丢失时以概率对 Burst 中的 TCP 进行显示拥塞指示,当下一个 Burst 丢失时,计算两个相邻的 Burst 丢失的时间间隔 t ,当 $t - T$ 大于某一阈值 Δt ,即 $t - T > \Delta t$ 时,以差值 ΔP 调整 $P: P \leftarrow P - \Delta P$;并用当前时间间隔 t 调整平均时间间隔 $T: T \leftarrow \gamma * T + (1 - \gamma) * t$ (其中 γ 表示修正因子,代表了旧的平均时间间隔 T 的权重,一般取接近 1 的一个值);当 $t - T < \Delta t$ 时,调整 $P: P \leftarrow P + \Delta P$;修正 $T: T \leftarrow \gamma * T + (1 - \gamma) * t$ 。提出以上计算概率 P 的方法是基于这样的事实:在 OBS 核心节点,由于信道竞争冲突而不是因为信道占用率过高的原因,Burst 会以一定的概率被随机丢弃,相邻被丢失 Burst 之间的时间间隔统计上是相等的,即平均时间间隔是不变的。如果平均时间间隔减小了,说明 Burst 的丢弃不仅仅是由于随机的竞争冲突,并且说明信道占用率正逐渐上升,此时应该增大平均标识概率 P ,相反的情况应该减小 P 。

伪代码如下:

```

public double cacul_p()
{
    double p, dert_p; //p 表示当前丢弃概率
                        //dert_p 表示调整因子
    int t, last_t, T, dert_t; //t 表示当前时间
                              //last_t 表示上一次 burst 丢失的时间
                              //T 表示平均间隔时间
                              //dert_t 表示时间阈值
    const double gama; //时间调整因子 gama
    boolean burst_lost = false; //标识 Burst 丢失
    initial(p); //初始化 P = P0
    initial(T); //初始化 T = T0
    initial(gama); //初始化 gama
    while(true && burst_lost == true) //当 Burst 丢失时, 进行调整
    {
        if(abs((t - last_t) - T) > dert_t)
        {
            p = p + sign((t - last_t) - T) * dert_p;
            t = gama * last_t + (1 - gama) * (t - last_t)
        }
        Burt_lost = false;
    }
    when(Burst 丢失)
        burst_lost = true;
}

```

(3) 当 Burst 到达 OBS 边缘节点 Egress 时, Egress 将其拆分成 IP 数据包向目的节点发送, 当数据包经过接收端的接入网络 RAN 中时, 网络中的路由器按着与 LAN 中的路由器相同的策略计算标记概率, 为了防止数据包在 LAN 和 RAN 中被重复标记, 在 RAN 中只是按着标记概率对数据包的 ECN 位进行取反操作即可, 这样在 LAN 中已经标记了的数据包的 ECN 位变成了 0 (因为在第一步中已经对这些数据包的发送源进行了拥塞通知), 对需要标记而未标记的数据包的 ECN 位赋 1。

2 仿真结果及结论

文中采用 Swarm 软件进行仿真, 下面首先对 Swarm 进行简单介绍: Swarm 的理论基础是复杂自适应系统 (CAS, Complex Adaptive System), 它是通过相对简单的微观个体活动可以突现出宏观层面的复杂行为, 这里微观个体的行为一般都比较简单, 而这些简单的微观的个体行为共同作用所表现出的宏观层面的行为比较复杂。Swarm 中最主要的四个部分, 往往也是一个 Swarm 模拟程序经常包括的四个部分是: 模型 swarm (ModelSwarm)、观察员 swarm (ObserverSwarm)、模拟主体和环境。模拟主体是用 Java 编写的普通 Jav-

aBean 类, 然后在 ModelSwarm 中统一调度这些主体的行为, 主体间通过环境相互作用, 最后在 ObserverSwarm 中观察仿真结果。Swarm 中的个体被成为 Agent, 本实验主要用到 TCP 发送端, TCP 接收端, 接入网络路由器, OBS 边缘节点和核心节点路由器等 Agent; 网络拓扑如图 3 所示。

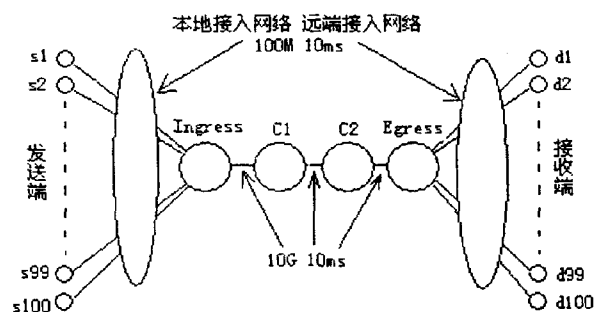


图 3 试验网络拓扑图

为了验证 Swarm 在 TCP 仿真方面的有效性, 类似文献[3]一样首先对 TCP 三种版本进行了仿真实验, 得到了相同的结论: 图中网络流量急剧下降处为 Burst 丢失的时刻, 由于 New-Reno 在数据包丢失时, 特别是在一个窗口内的全部数据丢失时, 每一 RTT 只重传一个丢失的 Packet, 性能最差, SACK 由于一次可以重传多个丢失的数据包, 性能最好, Reno 介于二者之间, 效果图如图 4 所示。更详细的内容请参考文献[3]。

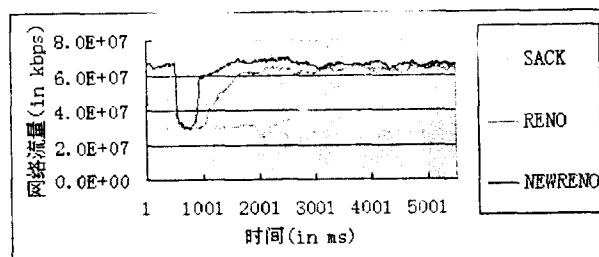


图 4 Burst 丢失时三种 TCP 的网络流量变化

下面采用 Swarm 对所提出的分段拥塞指示算法进行性能分析, 为简单起见, 采用第一种即根据信道占用情况来决定选择概率 P, 并假设 TCP 源端收到 ECN

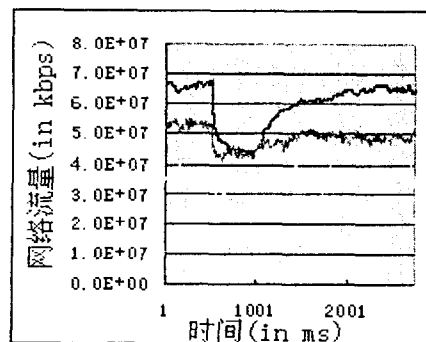


图 5 不同网络流量 Burst 丢失时网络流量的变化

(下转第 82 页)

本和增加透明的路由器的方法来模拟动态体系结构。

4 小结和展望

体系结构求精是 SA 相关研究内容中的一项重要工作。文中对几种典型的体系结构求精方法作了详细阐述,并对求精检测作了相关介绍。从研究现状来看,体系结构求精大都是在一种形式化方法上展开,而形式化方法不仅仅需要有形式化的描述语言和形式化的开发方法,还需要合适的模型检测工具支持。基于上述考虑,体系结构求精的重点在于寻找一种合适形式化方法,对体系结构模型进行形式化描述,提出切实可行的求精方法并制订求精规则,然后在规则的指导下逐步求精。最后对求精结果进行检测,验证其是否满足需求说明。因此,体系结构求精研究的根本在于精确地给出体系结构的行为求精规则。如何严格地、形式化地给出体系结构求精规则,特别是行为求精规则,成为以后研究工作的关键所在。

软件体系结构的动态演化是目前相关研究领域的一个热点问题。动态演化的一个核心问题是如何保证演化过程中的一致性和完整性。求精是软件开发中的基本活动,在体系结构的层次化设计过程中,求精可保证不同层次的体系结构信息维持一致和完整。故研究者也致力于研究如何通过体系结构求精进行软件体系结构动态演化的技术和机制。

参考文献:

- [1] 梅 宏,申峻嵘. 软件体系结构研究进展[J]. 软件学报, 2006,17(6):1257-1258.
- [2] 戎 玫,张广泉. 软件体系结构求精方法研究[J]. 计算机科学,2003,3(4):108-110.
- [3] Luckham DC, Vera J. An event-based architecture definition language[J]. IEEE Transactions on Software Engineer-

ing, 1995,21(9): 717-734.

- [4] Moriconi M, Qian X, Riemer Mchneider R. Correct Architecture Refinement[J]. IEEE Tran. Soft. Eng., 1995,21(4): 356-372.
- [5] Garlen D. Style-Based Refinement for Software Architecture [S]. ACM 0-89791-867-3. [s.l.]:[s.n.], 1996.
- [6] 张广泉. 基于 XYZ/E 的软件体系结构描述语言研究[J]. 计算机科学,2000,27(9,专辑):155-157.
- [7] 舒 明. 基于 XYZ/E 的软件体系结构描述和求精实例研究[D]. 北京:中国科学院,2001.
- [8] 晏荣杰,张广泉. 一种基于构件的软件体系结构求精方法及其应用[J]. 重庆师范学院学报:自然科学版,2003,6(7): 1-5.
- [9] 李长云. 基于体系结构的软件动态演化研究[D]. 杭州:浙江大学,2005.
- [10] 李长云,李贻生,何频捷. 一种形式化的动态体系结构描述语言[J]. 软件学报,2006,17(6):1349-1359.
- [11] Victor B, Moller F. The Mobility Workbench - A tool for the π -calculus[J]. 1994,2(2):6-10.
- [12] Kerschbaumer A. Behavioral Refinement of Software Architecture[D]. Graz: Technischen University Graz, 2002.
- [13] Allen R J. A Formal Approach to Software Architecture[R]. Technical Report CMU-CS-97-144. USA: Carnegie Mellon University, 1997.
- [14] Magee J, Dulay N, Eisenbach S, et al. Specifying Distributed Software Architectures[C]//In Proceedings of 5th European Software Engineering Conference. Spain: [s.n.], 1994:137-153.
- [15] Allen R J, Douence R, Garlan D. Specifying and Analyzing Dynamic Software Architectures [C]//In Proceedings of Foundations of Component-Based Systems Workshop. [s.l.]:[s.n.], 1997:21-37.
- [16] Mateescu R. Model checking for software architectures[C]//In Proceedings of the 1st European Workshop on Software Architecture. [s.l.]:Springer-Verlag, 2004:219-224.

(上接第 78 页)

指示时,采用与 RENO 相同的策略调整拥塞窗口。仿真结果如图 5 所示。

从图中可以看出,在网络流量比较大时,此时信道利用率比较高,Burst 的丢失不仅仅是因为偶然的竞争冲突。笔者选择了较大的标示概率 P,此时网络流量出现较大的下滑,有效地调节了网络的拥塞状况;当网络流量小时,Burst 的丢失偶然性较大,选择较小的概率,此时网络流量并没有出现较大的变化,网络保持稳定状态。

参考文献:

- [1] QIAO C, YOO M, Myungsik Y. Optical burst switching

(OBS): a new paradigm for an optical Internet[J]. Journal of High Speed Networks, 1999,8(1):69-84.

- [2] YU X, QIAO C. Performance evaluations of TCP traffic transmitted over OBS networks[R]. SUNY Buffalo: CSE Dept, 2003.
- [3] YU X, QIAO C, LIU Y. TCP implementations and false time out detection in OBS networks [C]//INFOCOM 2004, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies. New York: [s.n.], 2004:774-784.
- [4] Venkatesh T, Praveen K. Performance evaluation of high speed TCP over optical burst switching networks [J]. Optical Switching and Networking, 2007,4(1):44-57.