

控制时延的主动队列管理算法

吴 东*

(湛江师范学院 信息科学与技术学院, 广东 湛江 524048)

(* 通信作者电子邮箱 wdagx@126.com)

摘 要:针对现有的主动队列管理(AQM)算法造成的队列时延无法满足 VoIP、音视频等流媒体传输需求的问题,提出一种直接控制队列时延的主动队列管理算法——DCQA。该算法使用 PID 控制器计算路由器缓存的数据包丢弃概率,用其对即将进入缓存排队的数据包做丢包判断并采取相应动作,以控制队列时延在期望值以下。实验仿真了 3 种网络环境下 DCQA 的性能,链路利用率分别是 99.93%、99.88% 和 99.95%。并且,队列时延分别有 50.45%、51.59%、52.4% 被控制在期望值以下,比 CoDel 算法分别提高了 3.6%、40.53%、50.69%。实验结果表明,DCQA 在不同的网络环境中都可以获得较高的链路利用率,而且控制队列时延的能力优于 CoDel 算法,适用于流媒体的传输。

关键词:主动队列管理;缓存;队列;服务质量;时延

中图分类号: TP393.07 **文献标志码:** A

Active queue management algorithm of queue delay control

WU Dong*

(School of Information Science and Technology, Zhanjiang Normal College, Zhanjiang Guangdong 524048, China)

Abstract: To solve the problem that the queue delay cannot meet the demand of media applications, such as VoIP, real time video and remote video conference in the existing Active Queue Management (AQM) algorithms, a new AQM algorithm named DCQA (Direct Control Queue Delay Algorithm) was proposed. In this new algorithm, to control the queue delay below the expected value, identifying the packet dropping and other corresponding processes were done before a new packet entered the router buffer. The packet dropping rate was computed by PID controller. Then, a simulation in three network environments was conducted to compare the algorithm performance of DCQA and CoDel algorithm. The experimental results show that the queue delay can be effectively controlled and high utilization of link is obtained in DCQA. The detailed data of utilization of link are 99.93%, 99.88% and 99.95%. At the same time, the simulation results show that the probabilities of the queue delay below the expected value are 50.45%, 51.59% and 52.4%, improved by 3.6%, 40.53% and 50.69% compared with the CoDel algorithm. Therefore, DCQA is applicable for streaming media transmission.

Key words: Active Queue Management (AQM); buffer; queue; Quality of Service (QoS); delay

0 引言

为了解决计算机网络拥塞和提升计算机网络性能, Floyd 等^[1]于 1993 年建议在路由器中部署使用随机早期检测(Random Early Detection, RED)算法,从此拉开了主动队列管理(Active Queue Management, AQM)算法研究的序幕,并使其成为了计算机网络领域的研究热点之一。近年来,研究人员提出了大量的 AQM 算法,例如 RRED^[2]、ABlue^[3]、RQMQ^[4]等,它们采用不同的方法来管理路由器缓存队列,确实获得了很好的网络性能。但上述算法都把队列长度作为管理对象,要求队列长度收敛在期望值附近,该目标显然跟经过路由器的多媒体应用的目标不一致。例如,VoIP、音频、视频等网络应用都明确规定所能忍受的网络时延,而不是指出它们的数据包所经过的路由器的队列长度的期望值^[5]。这就让网络管理员在配置路由器时很难设定 AQM 算法的相关参数。并且,大容量缓存被广泛应用到当前的网络路由器中所造成的 Bufferbloat 问题^[6-7],也是上述 AQM 算法无法解决的。因此,

必须对现有的 AQM 算法进行改进。

2012 年, Nichols 等^[8]提出了 CoDel 算法,该算法直接对队列时延进行控制。当队列时延高于期望值时,路由器就进入丢弃状态不断地丢包,直到队列时延下降到期望值为止(使用 CoDel 算法会带来大的丢包率,但这是合理的,因为对于运行在端系统中的多媒体应用来说时延大的数据是无效的,所以当路由器发现数据包的时延已经高于期望值时就可以直接丢弃,无须再传给端系统)。实验结果表明,CoDel 算法确实能控制队列时延,满足了多媒体应用的需求。但深入研究发现,CoDel 算法为了计算时延需要给每个数据包设置一个时间标签,并且丢包操作是在数据包被接收到队列之后进行的,这些行为无疑会增加算法的工作量,而且很难在硬件中实现。

本文沿用 CoDel 算法直接控制队列时延的观点,提出了 DCQA(Direct Control Queue Delay Algorithm)。该算法使用 PID 控制器计算路由器缓存的数据包丢弃概率,在新的数据包进入缓存排队前做出丢包识别并采取相应动作,以获得期

收稿日期:2013-08-21;修回日期:2013-10-21。

基金项目:广东省自然科学基金资助项目(S2012010010438);湛江市科技攻关计划项目(2013B01148)。

作者简介:吴东(1981-),男,广西合浦人,讲师,硕士,CCF 会员,主要研究方向:网络拥塞控制。

望的队列时延。NS2^[9]仿真实验结果表明,DCQA 不仅可以使队列时延长时间低于期望值,还获得了较高的链路利用率。

1 直接控制队列时延算法——DCQA

DCQA 的基本思想是将路由器当前的队列时延与期望时延的差值输入到 PID 控制器中,计算出下一时刻的队列数据包的丢弃概率,并以此采取丢包动作,最终使队列时延低于预设的期望值。为了达到该目标,DCQA 需要解决 3 个关键问题:1) 队列时延的计算;2) 队列数据包丢弃概率的计算;3) 数据包的丢弃。下面详细介绍 DCQA 的具体实现,并分析其是如何解决上述问题的。

DCQA 的组成示意图如图 1 所示。该算法由一个丢弃模块(Drop Module, DM)、一个数据包队列(Packet Queue, PQ)、一个丢弃概率计算(Drop Probability Calculation, DPC)模块和一个出队列速率估计(Departure Rate Estimation, DRE)模块组成。其中 DRE 和 DPC 模块都是周期性工作的。当计时器发生超时, DRE 就重新计算出队列速率 dq_rate , 并将它传给 DPC。DPC 在获得 dq_rate 之后,联合 PQ 模块当前的队列长度 $qlen$ 计算队列时延,然后再使用 PID 控制器计算丢弃概率 $drop_prob$ 。DM 则根据 DPC 计算出的 $drop_prob$ 对新到达的数据包作丢包识别并采取相应的动作。

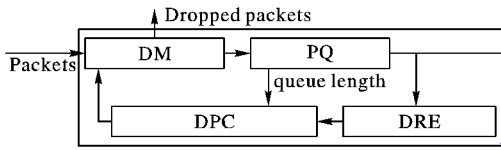


图1 DCQA 组成示意图

1.1 DRE 计算出队列速率的方法

考虑到网络负载会影响路由器的发送速度,因此不能简单地将数据包的出队列速率等同于链路的带宽。在 DCQA 中按照如下的方式计算出队列速率:算法开始工作时设定一个计时器,在计时时间内,当有数据包被发送出去后,使用式(1)统计这段时间内的数据发送总量;当计时器发生超时,根据式(2)计算当前的出队列速率;并且,为了有效地消除网络突发流量所引起的出队列速率的急速变化,在此引入了类似低通滤波器(low-pass filter)加权^[1]的方法,如式(3)计算出队列速率 dq_rate :

$$dq_count += dq_psize \quad (1)$$

$$cur_dq_rate = dq_count/T \quad (2)$$

$$dq_rate = (1-w) \times dq_rate + w \times cur_dq_rate \quad (3)$$

其中 T 是计时器的时间。

1.2 DPC 计算队列时延和丢弃概率的方法

CoDel 算法为入队列的每个数据包都设置一个时间标签,目的就是精确地计算每个数据包在路由器中的时延,但这是没有必要的。因为计算每个数据包的时延不仅增加运算量,而且根据这些时延来采取控制会过于频繁,将造成系统的不稳定,所以本文设计的 DCQA 只计算当前队列中的队尾数据包的时延。当计时器发生超时, DPC 模块根据式(4)计算当前的队列时延:

$$cur_del = qlen/dq_rate \quad (4)$$

其中, dq_rate 是 DRE 模块计算得到的出队列速率。然后再使用 PID 控制器计算丢弃概率,如式(5)所示:

$$drop_prob += K_i * (cur_del - ref_del) + K_p * (cur_del -$$

$$pre_del) + K_d * (cur_del - 2 * pre_del + last_del) \quad (5)$$

其中: ref_del 是期望队列时延; cur_del 是当前队列时延; pre_del 、 $last_del$ 是前两次的队列时延;而 K_p 、 K_i 、 K_d 分别是 PID 控制器的比例、积分和微分常数。

1.3 DM 丢弃数据包的方法

如何丢包是 DCQA 要解决的核心问题之一,在 DM 中采用阈值法设计了两种不同的丢包策略。当路由器收到数据包时,按照式(6)来处理数据包:

$$\begin{cases} \text{drop,} & qlen > q_{max} \\ \text{randomly drop with probability } drop_prob, & qlen \leq q_{max} \end{cases} \quad (6)$$

其中: $qlen$ 是当前队列长度, q_{max} 是路由器允许的最大队列长度, $drop_prob$ 是当前的丢弃概率。从式(6)可以看出, q_{max} 是影响丢包的关键,因此要合理设定。DM 根据式(7)计算 q_{max} :

$$q_{max} = ref_del * link_bandwidth \quad (7)$$

其中: ref_del 是期望队列时延, $link_bandwidth$ 是路由器的出口链路带宽。因为这两个参数与网络环境、应用需求有关,所以 DCQA 能在各种网络中运行,并取得很好的效果。

1.4 算法流程

DCQA 的具体实现伪代码如下:

```
1) Initialize;
2) start_timer(T);
3) For each arrival packet:
4)   if (qlen > q_max) then
5)     drop the arrival packet;
6)   else if (rand() < drop_prob) then
7)     drop the arrival packet;
8)   else {
9)     enqueue the arrival packet;
10)    qlen += packet_size; }
11) For each departure packet:
12)   qlen -= packet_size;
13)   dq_count += packet_size;
14)   dequeue the departure packet;
15) For each Timer expired:
16)   cur_dq_rate = dq_count/T;
17)   dq_rate = (1-w) * dq_rate + w * cur_dq_rate;
18)   cur_del = qlen/dq_rate;
19)   drop_prob += K_i * (cur_del - ref_del) + K_p * (cur_del -
20)     pre_del) + K_d * (cur_del - 2 * pre_del + last_del)
21)   last_del = pre_del;
22)   pre_del = cur_del;
23)   dq_count = 0;
24)   start_timer(T);
```

伪代码中的第 2) 行是算法开始执行时启动计算丢弃概率的计时器;第 4) ~ 10) 行是对新到达的数据包进行丢包判断并采取相应的动作;第 12) ~ 14) 行在发送数据包后更新队列长度和计时时间内的发送数据量;第 16) ~ 23) 行计算丢弃概率,将发送数据量清零,并重启计时器。

2 实验结果与性能分析

为了评估 DCQA 的性能,本文在 NS2 网络仿真实验平台上开展了一系列的实验,并与 CoDel 算法进行了比较。实验使用如图 2 所示的网络拓扑结构。其中: $S_1 \sim S_n$ 是发送端, $R_1 \sim R_n$ 是接收端,中间两个节点 N_1 和 N_2 是路由器;路由器

与端节点间的链路带宽都为 100 Mb/s,传输时延为 5 ms,队列算法为 DropTail;路由器之间的链路带宽为 10 Mb/s,传输时延为 40 ms,队列算法为 DCQA 或 CoDel;路由器的缓存大小为 350 packet。并且,DCQA 的参数 $K_p = 0.125$ 、 $K_i = 1.25$ 、 $K_d = 0.0125$,期望队列时延 ref_del 设定为 30 ms,路由器允许的最大队列长度 q_{max} 是 300 packet,计时器的时间为 30 ms。CoDel 算法的期望队列时延也是 30 ms,其他参数使用文献[8]的默认值。在此进行了代表不同网络环境的 3 次仿真实验,并使用队列时延的累积百分比来对比 DCQA 和 CoDel 算法控制队列时延的性能。

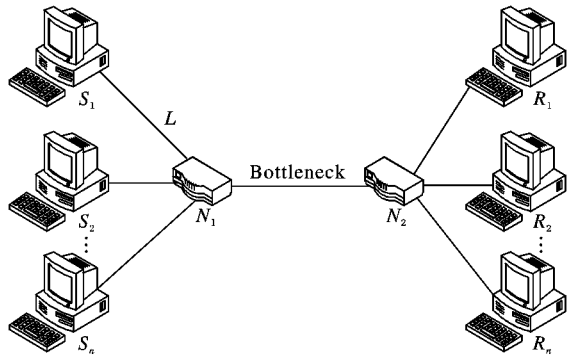


图2 实验用的网络拓扑结构

实验1 分析算法在轻负载情况下的性能。在实验开始后的 1 s 内共随机启动 50 个 TCP 流来发送数据,整个实验过程持续 100 s。仿真结果如图 3 所示。从图 3 中可以得知,DCQA 在 30 ms 处的累积百分比大于 CoDel 算法的(DCQA 的是 50.45%,CoDel 算法的是 46.85%);从而说明使用 DCQA 时,能获得更好的队列时延,有 50.45% 的时间小于期望的 30 ms。

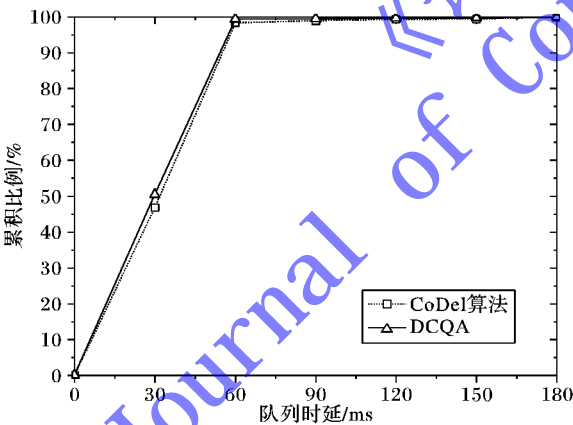


图3 实验1的队列时延的累积百分比

实验2 分析算法在重负载情况下的性能。在实验开始后的 1 s 内共随机启动 100 个 TCP 流来发送数据,整个实验过程持续 100 s。仿真结果如图 4 所示。从图 4 中可以得知,DCQA 在 30 ms 处的累积百分比要大于 CoDel 算法的(DCQA 的是 51.59%,CoDel 算法的是 11.06%);从而说明使用 DCQA 时,有 51.59% 的时间其队列时延是小于期望的 30 ms,而 CoDel 算法只有 11.06%。

实验3 分析算法在混合业务流情况下的性能。具体的实验过程是:在实验开始后的 1 s 内共随机启动 5 个 TCP 流和 2 个 UDP 流来发送数据,整个实验过程持续 100 s。仿真结果如图 5 所示。从图 5 中可以得知,DCQA 在 30 ms 处的累积

百分比要远大于 CoDel 算法的(DCQA 的是 52.4%,CoDel 算法的是 1.71%)。统计数据说明使用 DCQA 能获得更好的队列时延。

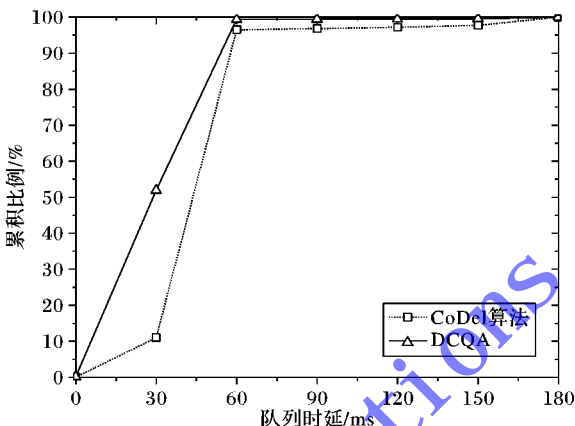


图4 实验2的队列时延的累积百分比

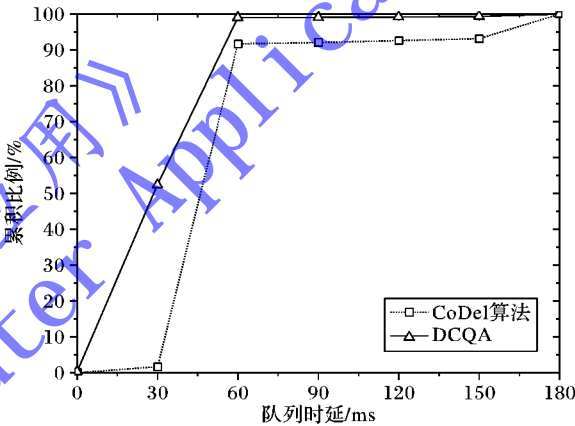


图5 实验3的队列时延的累积百分比

综合以上 3 次实验的结果,可以得知 DCQA 与 CoDel 算法相比能更好地控制队列时延。在网络轻负载情况下 DCQA 比 CoDel 算法的累积百分比高 3.6%,而在网络重负载、存在混合业务流情况下更是分别高出 40.53% 和 50.69%。这说明 DCQA 更适合在现今的多媒体网络^[10]以及即将来临的大数据^[11-12]背景下的网络中使用。

另一方面,统计了使用 DCQA 所获得的链路利用率。从表 1 的数据可以得知,网络中的 Bottleneck 链路的利用率都很高。这说明,虽然 DCQA 是通过丢弃数据包来达到控制队列时延的目的,但并没有影响网络中 Bottleneck 链路的利用率。

表1 DCQA 的链路利用率

实验序号	实验特征	链路利用率/%
1	轻负载	99.93
2	重负载	99.88
3	混合业务流	99.95

3 结语

本文提出了一种直接控制队列时延的主动队列管理算法——DCQA,在不同的网络环境中获得了较好的队列时延和较高的链路利用率,为多媒体应用的数据传输提供了服务质量(Quality of Service, QoS)保证;并且,算法的目标参数是

(下转第 639 页)

- 53(2): 217-230.
- [2] NIE Y. Research on CAPWAP protocol system implementation technology [D]. Wuhan: Huazhong University of Science and Technology, 2011. (聂勇. CAPWAP 协议系统实现技术研究[D]. 武汉: 华中科技大学, 2011.)
 - [3] ZHANG G. CAPWAP protocol and its application in integrated switch [D]. Xi'an: Xidian University, 2009. (张弓长. CAPWAP 协议及其在有线无线一体化交换机中的应用[D]. 西安: 西安电子科技大学, 2009.)
 - [4] GUO Z. Design and implementation of wireless access point extensions based on CAPWAP protocol [D]. Xi'an: Xidian University, 2011. (郭子明. 基于 CAPWAP 协议的无线接入点扩展的设计与实现[D]. 西安: 西安电子科技大学, 2011.)
 - [5] CALHOUN P, MONTEMURRO M, STANLEY D. RFC 5415, Control And Provisioning of Wireless Access Points (CAPWAP) protocol specification [S]. Fremont, CA: IETF, 2009.
 - [6] CALHOUN P, MONTEMURRO M, STANLEY D. RFC 5416, CAPWAP protocol binding for IEEE 802.11 [S]. Fremont, CA: IETF, 2009.
 - [7] China Mobile Communications Corporation. China mobile WLAN AC-AP interface interoperability test specifications: basic protocol volume (Version: 1.0.5) [M]. Beijing: China Mobile Communications Corporation, 2011. (中国移动通信集团公司. 中国移动 WLAN AC-AP 接口互通测试规范(基本协议分册)(版本号: 1.0.5) [M]. 北京: 中国移动通信集团公司, 2011.)
 - [8] RESCORLA E, MODADUGU N. RFC 4347, datagram transport layer security [S]. Fremont, CA: IETF, 2006.
 - [9] LIANG W, DONG P, ZHANG H. Research and implementation of centralized management and configuration based on CAPWAP [J]. Compute Technology and Development, 2011, 22(11): 69-72. (梁文波, 董平, 张宏科. 基于 CAPWAP 的集中配置管理机制的研究与实现[J]. 计算机技术与发展, 2012, 22(11): 69-72.)
 - [10] HUANG J, DAI Q, JIANG W. Implementation of centralized WLAN based on CAPWAP [J]. Mobile Communications, 2011, 35(8): 68-72. (黄军记, 戴青云, 姜文超. 基于 CAPWAP 的集中式 WLAN 实现[J]. 移动通信, 2011, 35(8): 68-72.)
 - [11] HUANG J. Improvement and implement the CAPWAP protocol on the fit AP system [D]. Guangzhou: Guangdong University of Technology, 2011. (黄军记. 瘦 AP 系统中 CAPWAP 协议的改进与实现[D]. 广州: 广东工业大学, 2011.)
 - [12] BERNASCHI M, CACACE F, DAVOLI A, et al. A CAPWAP-based solution for frequency planning in large scale networks of WiFi hot-spots [J]. Computer Communications, 2011, 34(11): 1283-1293.
 - [13] XIANG W, WANG Z, GAO C. Communication protocol of centralized WLAN architecture [J]. Computer Engineering, 2008, 34(22): 115-117. (向望, 王志伟, 高传善. 集中式 WLAN 体系结构通信协议[J]. 计算机工程, 2008, 34(22): 115-117.)
 - [14] MENG Y. Centralized wireless LAN architecture based on CAPWAP protocol [J]. Network Security Technology and Application, 2009(9): 29-31. (孟一飞. 基于 CAPWAP 协议的集中式无线局域网架构[J]. 网络安全技术与应用, 2009(9): 29-31.)
 - [15] XIONG P, GUO J, ZHANG G. Design and implementation of CAPWAP-based AP simulator [J]. Study on Optical Communications, 2012(6): 64-66. (熊鹏, 郭见兵, 张光华. 基于 CAPWAP 协议的 AP 模拟器的设计与实现[J]. 光通信研究, 2012(6): 64-66.)

(上接第 634 页)

多媒体应用的时延需求,让管理员使用时更容易配置,这将为算法的推广带来了有利条件。但 DCQA 是单队列机制,还无法为不同的业务流设定不同的期望时延,将在下一步工作中将对其进行改进。

参考文献:

- [1] FLOYD S, JACOBSON V. Random early detection gateways for congestion avoidance [J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413.
- [2] ZHANG CW, YIN J P, CAI Z P, et al. RRED: robust RED algorithm to counter low-rate denial-of-service attacks [J]. IEEE Communications Letters, 2010, 14(5): 489-491.
- [3] FENG W, CHEN Y, HU Y, et al. Adaptive active queue management algorithm ABlue [J]. Computer Engineering, 2011, 37(10): 84-86. (冯伟, 陈元琰, 胡愚, 等. 一种自适应主动队列管理算法 ABlue[J]. 计算机工程, 2011, 37(10): 84-86.)
- [4] WANG J, LIN B. Active queue management algorithm based on particle swarm optimization [J]. Journal of Computer Applications, 2013, 33(2): 390-392. (王军祥, 林柏钢. 基于粒子群优化的主动队列管理方法[J]. 计算机应用, 2013, 33(2): 390-392.)
- [5] H3C. 从应用的角度评估 DDoS 防护的性能 [EB/OL]. [2013-06-03]. http://www.h3c.com.cn/Products_Technology/Products/IP_Security/Home/Technology_Doc/200908/645270_30003_0.htm.
- [6] GETTYS J. Bufferbloat: dark buffers in the Internet [J]. IEEE Internet Computing, 2011, 15(3): 95-96.
- [7] GETTYS J, NICHOLS K. Bufferbloat: dark buffers in the Internet [J]. Communications of the ACM, 2011, 55(1): 57-65.
- [8] NICHOLS K, JACOBSON V. Controlling queue delay [J]. Communications of the ACM, 2012, 55(7): 42-50.
- [9] The network simulator — NS2 [EB/OL]. [2013-06-03]. <http://www.isi.edu/nsnam/ns/>.
- [10] YIN H, ZHAN T, LIN C. Multimedia networking: from content delivery networks to future Internet [J]. Chinese Journal of Computers, 2012, 35(6): 1120-1130. (尹浩, 詹同宇, 林闯. 多媒体网络: 从内容分发网络到未来互联网[J]. 计算机学报, 2012, 35(6): 1120-1130.)
- [11] MEN X, CI X. Big data management: concepts, techniques and challenges [J]. Journal of Computer Research and Development, 2013, 50(1): 146-169. (孟小峰, 慈祥. 大数据管理: 概念、技术与挑战[J]. 计算机研究与发展, 2013, 50(1): 146-169.)
- [12] WU H. The challenges of network technology and industry based on big data [J]. Communications World Weekly, 2013(4): 31. (邹贺铨. 大数据对网络技术和产业的挑战[J]. 通信世界, 2013(4): 31.)