文章编号:1000 - 582X(2005)12 - 0055 - 04

基于 TTL 拥塞控制的主动队列管理算法*

吕建斌[1],廖晓峰[2]
(重庆大学 1. 图书馆; 2. 计算机学院,重庆 400030)

摘 要:提出了一种新颖的主动队列管理算法——基于 TTL(Time to Live)的 ECN 及 BECN 的综合. ECN 和 BECN 在指示拥塞的过程中各有优缺点,二者的综合可望提高拥塞指示的效率. TTL 是在网络上传输的分组必须具有的属性,且每一次转发都要经过检测,以决定该分组的处理方式——转发或丢弃. 通过对 TTL 的判断来决定网络拥塞指示的方式——ECN 或 BECN. 建立了一个数学模型,对模型的分析结果表明该算法对控制拥塞、提高网络吞吐量等有更好的效果. 在 NS 环境下对算法进行了仿真,仿真结果支持了理论分析.

关键词:随机早期检测;显式拥塞通告;后向显式拥塞通告;生存期;主动队列管理
中图分类号:TP393 **文献标识码**:A

近年来主动队列管理算法(AQM)[1]被广泛地用于拥塞控制. 在 AQM 中,网络中的路由节点通过对其维护的分组队列的长度进行检测,以提前检测拥塞的出现并通知发送方,这样发送方可以在更严重的拥塞发生之前启动拥塞避免算法以缓解拥塞. 同时为了更加有效地进行拥塞指示,显式拥塞指示(ECN)以及反向显式拥塞指示(BECN)等拥塞指示方式被提出且在深入的研究中.

1 随机早期检测(RED)算法以及 ECN/BECN

随机早期检测(RED)[2]算法是一种得到广泛研究的主动队列管理算法,它通过在队列未满之前丢包来达到控制拥塞的目的. 与 Drop Tail 相比,RED 为队列管理增添了 2 种新机制:1)不是等队列全满后再丢弃到来的分组,而是利用概率判定机制事先丢弃部分分组来预防可能发生的更加严重的拥塞;2)通过平均队列长度而非实时队列长度调整分组丢失概率,由此来尽可能地吸收部分短暂的突发流量. 该算法利用 EWMA 来计算平均队列长度,它为平均队列长度设定了 2 个阈值 min_{th} 和 max_{th},如果平均队列长度小于 min_{th},则没有分组被丢弃;如果平均队列长度大于 max_{th},则所有到达的分组都将被丢弃;如果平均队列介于 min_{th} 和 max_{th} 之间,则以一定的概率丢弃分组,这个概率是平均队列长度的函数.

RED 在有效控制平均队列长度、适应突发数据流等方面有相当的改进,但仍然存在一些缺陷,如:RED 算法的性能对参数相当敏感,在特定的网络负载状况下依然会导致多个 TCP 的同步,造成队列震荡、吞吐量降低和时延抖动加剧等. 其公平性和稳定性也存在问题. 同时其通过分组的丢失来隐式地指示拥塞,所产生的延迟在链路负载很重的情况下对拥塞指示的影响是很大的.

显式拥塞指示 ECN(Explicit congestion notification)[3]是对 AQM 的一个补充,通过对分组进行标记而不是丢弃来显式地通知源端可能或正在发生的拥塞. AQM 以某种策略选择了一个分组后,ECN 标记其 CE(Congestion experiencing)标志位,然后将其继续转发. 当被标记的分组到达接收端后,接收端在应答的 ACK 中标记其 CE 位. 源端在接收到被标记了的 ACK 后,将拥塞窗口(congestion window)减半,同时将下一个要发送的分组头中的 CWR(Congestion Window Reduced)位置位. 接收端继续将 ECN-Echo 置位,直到接收到标记了 CWR 位的分组.

ECN 的优点体现在 2 个方面:1)避免对短的或对延迟敏感的 TCP 连接的分组的不必要的丢弃;2)避免不必要的超时重传. 但 ECN 同时有一个明显的缺陷:当网络的延迟比较大时,拥塞指示将会产生一个明显的滞后,因而导致网络性能的大幅振荡.

后向显式拥塞指示 BECN(Backward-ECN)[4-6]是另一种显式的拥塞指示方式. 不同于 ECN 的反馈方式,BECN 通过在拥塞节点产生 ISQ(ICMP Source Quench)分组来显式地指示拥塞. 同 ECN 一样,BECN 同样是对 AQM 的补充. 当 AQM 检测到队列长度达到某个预先设定的值时,不是选择传统的丢弃分组的方

* 收稿日期:2005 - 07 - 05
作者简介:吕建斌(1976 -),男,内蒙古呼和浩特人,重庆大学图书馆馆员,硕士,主要从事网络安全、拥塞控制等研究.

式来指示拥塞,而是将该分组标记转发(或者丢弃),同时发送一个 ICMP 源抑制(ISQ)分组到标记分组的源端,源端在接收到 ISQ 分组后,采取与 ECN 源端同样的方式来响应. 而已经被标记转发的分组则不会触发下一个支持 BECN 的路由器继续发送 ISQ 分组到同一个发送方. TCP 发送方不需要初始化以支持 BECN. 在接收到 ISQ 后,发送方置 CW 位,并且将 ssthresh 设置为当前拥塞窗口的一半,直到一个 RTT 后再开始增加拥塞窗口. 发送方在一个 RTT 周期内只响应一次 ISQ.

BECN 在解决迟延的方面比 ECN 更加有效,因而可以使网络运行更加平稳. 但它会在一定程度上加重反向链路的负载. 同时 ISQ 在 ICMP 分组处理方面会给网关增加更多的负担.

2 基于 TTL 的拥塞控制

基于上述的原因,笔者提出了一种既可以缩短时延又不会对反向链路造成很大压力的算法,称之为基于 TTL 的主动队列管理算法 TRED(TTL Based RED). TTL(生存期)是指分组的最大存在时间,由 Internet 层首部的一个字节表示. 分组每通过一个路由器,其 TTL 值就减少 1,如果 TTL 值减为 0,则丢弃该分组,并发送一个 ICMP 分组给最初的发送者. 算法利用分组的这个特性,对分组进行分类,采取不同的方式进行处理.

在采用 RED 进行拥塞控制的网络中,若在某路由节点上检测到拥塞,同时到达的分组的 TTL 比较大,则说明该分组距离源端比较近(逻辑上,亦即转发的次数少),从统计的角度而言离接收端比较远,若采用 ECN 方式的话,则拥塞指示的滞后是很大的,此时算法采用 BECN 的反馈方式,发送 ISQ 给源端来通知发送方及时调整其发送速率. ISQ 分组可以通过很少的几次转发快速到达源端,这样对反向链路不会造成很大的压力,发送方可以及时地运行拥塞避免策略. 反之,若 TTL 比较小,说明分组已经传输了很久(经历了很多次转发),那么算法假定该分组很快就要到达接收方,此时算法就采取 ECN 的方式,对它进行标记. 算法设置一个参数 TH_{tl},当 TTL 大于 TH_{tl} 的时候,算法采用 BECN 机制显式地通知源端;反之,则采用 ECN 的方式来进行拥塞指示.

对于某一个单一的连接,按照 S. Floyd 在文献[7]中给出的公式,其最大吞吐量为

$$T \leqslant \frac{1.5\sqrt{2/3} * B}{R * \sqrt{p}}, \tag{1}$$

式(1)表示一个单一连接从拥塞避免到达拥塞窗口 W 这一阶段,亦即 $W/2$ 个 RTT 时间之中,可达到的最大吞吐量,其中 B 是分组大小(缺省值为 512 byte),R 是一个 RTT(包括排队时延),p 是平均丢包率. 该公式描述了一个连接以 $W/2$ 个 RTT 为周期进行的周期性的数据传输.

当不考虑除拥塞节点外的其他中间节点对分组的丢弃的时候,该连接在这个周期之内的有效吞吐量 G(GOODPUT)与 T 是相等的,即有

$$G = T \leqslant \frac{1.5\sqrt{2/3} * B}{R * \sqrt{p}}. \tag{2}$$

在一个给定的网络中,只有一个瓶颈路由器 B,讨论所有通过 B 到达一个固定接收节点 D 的一个子网 C 的吞吐量情况(如图 1,其中节点 7 是拥塞路由器 B,节点 25 是终端节点 D).

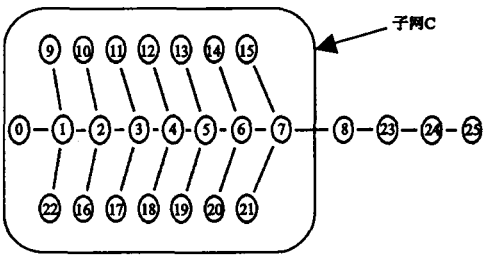

图 1 拓扑结构

在该子网中,只有叶子节点,亦即终端节点对整个子网 C 的吞吐量是有贡献的. 定义任一与路由器 B 之间的节点数目为 i(实际上,i 是 TTL 相对于最大允许 TTL 值的补)的终端节点为 N_{ij},其中 j 表示有其他节点与 B 之间的节点数目也等于 i. 为方便讨论,假定任意相邻两个节点之间的时延是相等且恒定的,用 m 表示. 路由器 B 到接收方 D 之间的节点数目为 d. 则任一节点 N_{ij} 的一个 RTT 为 $(2(i+d)m+q)$,其中 q 就是在文献[7]中提到的排队时延,出于同样的原因,这里只考虑在拥塞节点 B 的排队时延. 同时,文献假定任一连接在发送窗口达到 W 的时候,最后一个分组且仅有最后一个分组被 B 标记. 假定,发送窗口达到 W 后,若发送窗口不立即减小,则所有后来的分组都将被丢弃. 如果网络中节点全部响应 ECN 的拥塞指示方式,则该被标记的分组需要一个 RTT 才能够到达源端,因此在该 RTT 中的分组将全部被丢弃. 在 ECN 作为拥塞指示的一个连接中,数据传输是以 $(W/2+1)$ 个 RTT 为周期进行的,且每个周期之中的有效吞吐量就是式(2)中的 G. 在给定时间 T 中,任一支持 ECN 的节点 N_{ij} 的有效吞吐量 G_{ijE} 可以表示为

$$G_{ijE} \leqslant \frac{T}{2(i+d)m+q} * \frac{1}{W/2+1} * \frac{1.5\sqrt{2/3} * B}{R * \sqrt{p}}. \tag{3}$$

若子网 C 中的所有节点采用 ECN 方式进行拥塞指示,则整个子网 C 的有效吞吐量可以表示为

$$G_E \leqslant \sum_C \frac{T}{2(i+d)m+q} * \frac{1}{W/2+1} * \frac{1.5\sqrt{2/3} * B}{R * \sqrt{p}}. \tag{4}$$

若网络中的所有节点都采用的是 BECN 的拥塞指示方式,那么进行拥塞指示的 ISQ 分组将在小于一个 RTT 的时间之内到达源端. 对于任一节点 N_{ij},当其发

送窗口达到 W 的时候,同样的最后一个分组被 B 标记,亦即该分组被丢弃,同时 B 产生一个 ISQ 分组发送到 N_{ij}. ISQ 到达 N_{ij} 的时延为 $2i$,则该连接是以 $(W/2+i/(i+d))$ 个 RTT 为周期进行传输的. 同时,由于 ISQ 会对反向链路造成压力,笔者假定该 ISQ 分组在每一次转发的过程中导致其它连接一个分组的丢失. 由于节点是任意的,为了讨论方便,认为丢失的分组都是属于该节点的. 那么,给定时间内 N_{ij} 的有效吞吐量 G_{ijE} 可以表示为

$$G_{ijB} \leqslant \frac{T}{2(i+d)m+q} * \frac{1}{\frac{W}{2}+\frac{i}{i+d}} * \left(\frac{1.5\sqrt{2/3}*B}{R*\sqrt{p}} - 1 - i\right). \tag{5}$$

在采用 BECN 作为拥塞指示的网络中,子网 C 的有效吞吐量为

$$G_B \leqslant \sum_{C} \frac{T}{2(i+d)m+q} * \frac{1}{\frac{W}{2}+\frac{i}{i+d}} * \left(\frac{1.5\sqrt{2/3}*B}{R*\sqrt{p}} - 1 - i\right). \tag{6}$$

比较式(3)和式(5)可知,任意单一连接在时间 T 内的吞吐量是受很多因素影响的,这里只考虑 i 对吞吐量的影响. 式(3)和式(5)都是 i 的减函数,显然式(5)的下降速度更快,因而通过它们的示意函数(图 2)可知,两者之间必然有一个交点,这就是算法要找的阈值 TH_{ttl}. 在 i 小于该阈值时,算法选择 BECN;反之,则采用 ECN.

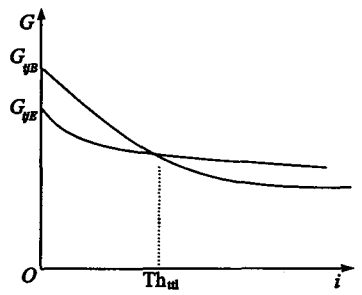

图 2　有效吞吐量

联立式(3)和式(5),在考虑最优化(即两者都取等号)的情况下有

$$\frac{1}{W/2+1} * \frac{1.5\sqrt{2/3}*B}{R*\sqrt{p}} = \frac{1}{\frac{W}{2}+\frac{i}{i+d}} * \left(\frac{1.5\sqrt{2/3}*B}{R*\sqrt{p}} - 1 - i\right), \tag{7}$$

$$i = \frac{\sqrt{(d-1)^2 + \frac{6BW}{(w+2)*R}*d} - (d+1)}{2}. \tag{8}$$

该值求出的就是 TH_{ttl},子网 C 的吞吐量 G 表示为

$$G = \sum_{C_1} \frac{T}{2(i+d)m+q} * \frac{1}{W/2+1} * \frac{1.5\sqrt{2/3}*B}{R*\sqrt{p}} + \sum_{C_2} \frac{T}{2(i+d)m+q} * \frac{1}{\frac{W}{2}+\frac{i}{i+d}} * \left(\frac{1.5\sqrt{2/3}*B}{R*\sqrt{p}} - 1 - i\right), \tag{9}$$

C_1 和 C_2 分别表示采用 ECN 和 BECN 作为拥塞指示的节点集合.

3　仿真及分析

笔者采用 NS-2[8] 作为模拟软件. 网络拓扑结构如图 1. 在该算法的模拟中,节点 7 和节点 8 之间的链路是瓶颈链路. 节点 0 以及节点 9-22 是发送方,节点 1-8 以及节点 23、24 是路由节点,节点 25 是惟一接收方. 在模拟中,节点 7 和节点 8 之间的网络带宽和延迟分别是 5 M 和 10 ms,其它各节点之间的带宽和延迟为 50 M 和 2 ms. 发送方和接收方分别采用 TCP/Reno 和 TCPSink 来支持端系统的拥塞响应. 其中发送方均发送 FTP 数据包,参数设置方面: $max_{th}=30$, $min_{th}=10$,其他参数均采用了系统提供的缺省值. 模拟时间为 20 s.

阈值 TH_{ttl} 的确定采取了如下的分析: $d=3$, $B=512$,设丢包率为 0.01,则与 W 的关系可得 $W=15$, R 取平均 RTT 为 42(此处忽略了排队时延),代入式(8)可得 $i=5$. 模拟环境中 TTL 的缺省值是 32,因此阈值 TH_{ttl} 设置为 27.

图 3 显示的是平均队列长度,TRED 在控制平均队列长度方面比另外两种机制更加稳定,且多数情况下要比另外两种机制下的平均队列长度小. RED/ECN 的平均队列长度虽然也在 20 左右,但其波动要大得多;而 RED 的波动最大.

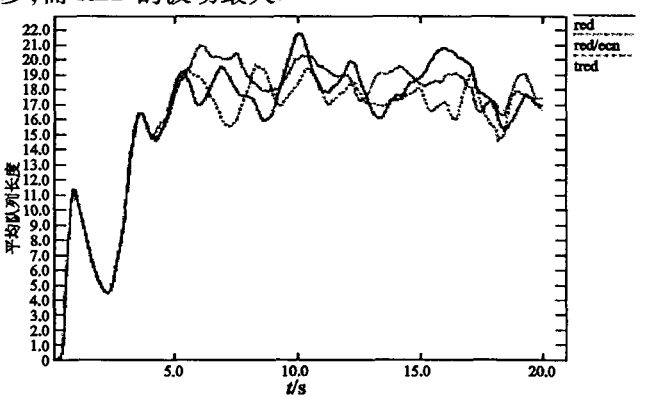

图 3　不同时间的平均队列长度

图 4 反应的是应用 3 种主动队列管理算法的有效吞吐量(goodput). 有效吞吐量是指被接收端接收到的分组数量. 图中最后的 0 表明传输结束. 从图 4 中可以发现 TRED 的有效吞吐量是优于另外二者的,尤其在 1 s 和 2 s 有新的负载加入到网络中的时候,RED 和

RED/ECN 都出现了明显的抖动,而 TRED 的抖动则要小得多,表现了很好的稳定性以及瓶颈链路利用率.

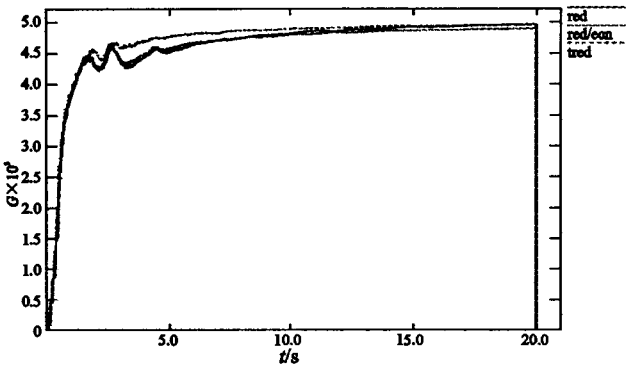

图 4 不同时间的有效吞吐量

4 结 语

对基于主动队列管理的拥塞控制算法——RED, RED/ECN 以及 RED/BECN 进行了详细的介绍,指出了当前主动队列管理算法都面临的一个问题,就是如何及时有效地通知源端当前的网络拥塞. 为了更好地进行网络拥塞的指示,提出了一个新的算法——TRED. 通过一个简化的数学模型对算法进行了分析,给出了算法可行的依据. 仿真结果表明 TRED 在控制路由节点上的队列长度、提高网络有效吞吐量、缩短端到端延迟以及降低丢包率等方面都要优于文中提到的几种主动队列管理算法.

当然,算法研究的是在网络负载较大但相对稳定的情况下所表现出来的性能,在网络负载变化较大的情况下算法的性能将是下一步研究的重点. 同时,如何选取参数以及整个网络的公平性研究也必然是未来的一个研究方向.

参考文献:

[1] BRADEN B, CLARK D, CROWCROFT J, et al. Recommendations on Queue Management and Congestion Avoidance in the Internet[DB/OL]. RFC2309, 1998.

[2] FLOYD S, JACOBSON V. Random Early Detection Gateways for Congestion Avoidance[J]. IEEE/ACM Transactions on Networking, 1993, 1(4):397-413.

[3] FLOYD S, RAMAKRISHNAN K. A Proposal to add Explicit Congestion Notification (ECN) to IP[DB/OL]. RFC 2481, 1999.

[4] HADI SALIM J, NANDY B, SEDDIGH N. A Proposal for Backward ECN for the Internet Protocol (IPv4/IPv6) [DB/OL]. IETF Draft Draft-jhsbnns-ecn-00.txt, June 1998.

[5] FRANK AKUJOBI, IOANNIS LAMBADARIS, RUPINDER MAKKER, et al. BECN for Congestion Control in TCP/IP Networks: Study and Comparative Evaluation [DB/OL]. http://www.sce.carleton.ca/faculty/lambadaris/recent-papers/globecom2002.pdf, 2002.

[6] PENG FEI, VICTOR LEUNG C M. Fast Backward Congestion Notification Mechanism for TCP Congestion Control[DB/OL]. Proc. IEEE IPCCC'02, Phoenix, AZ, 2002.

[7] NS. UCB/LBNL/VINT Network Simulator[CP/OL]. http://www.isi.edu/nsnam/ns/index.html, 2004.

[8] SALLY FLOYD, KEVIN FALL. Promoting the Use of End-to-End Congestion Control in the Internet[EB/OL]. http://www.icir.org/floyd/papers/collapse.feb98.pdf, 1998.

New AQM Algorithm: TTL based Congestion Control

LV Jian-bin[1], *LIAO Xiao-feng*[2]

(1. Library; 2. College of Computer Science, Chongqing University, Chongqing 400030, China)

Abstract: This paper proposes a novel algorithm——combination of ECN and BECN based on the value of the TTL (time to live)——the hops that a packet has been ever retransferred. Since both ECN and BECN have advantages and disadvantages, the combination may to enhance the effectiveness of the congestion indication. A default attribute of a data packet transmitted on a network——the TTL must be checked on each inner node (such as router or switcher) to decide whether to drop or forward. Based on the value of the TTL, ECN or BECN will be selected to inform the congestion condition to the sender, which will react to the indication. The mechanism can exploit the advantages of both ECN and BECN, and will not worsen the reverse link heavily. The mathematical model of TRED is constructed. Based on this model, the feasibility and the effectiveness of this algorithm are carefully discussed. The analysis results show that it does better than previous algorithm in controlling the congestion and improving the throughput in a congested network. The simulation results show measurable improvement in both queue length and throughput.

Key words: RED; ECN; BECN; TTL; AQM

(编辑 张 苹)