

基于 ARED 排队算法的定性探讨^{*}

刘晏兵 刘 蕾

(重庆邮电学院计算机系 重庆400065)

摘 要 队列管理和拥塞避免机制是实现网络 IP 服务质量(QoS)的保证。ARED 是一种拥塞避免技术,它在平均队列长度超过最小阈值时,以非零几率丢弃数据分组,尽量不使网络到达拥塞状态,从而克服了全局同步现象,提高了网络性能。本文对 ARED 算法及参数进行了定性分析,这对组建未来通信网路由器具有参考的作用。

关键词 拥塞,自适应 RED,排队算法

The Qualitative Study of ARED Queue Algorithms in Network

LIU Yan-Bing LIU Lei

(Department of Computer Science,Chongqing University of Post and Telecommunications,Chongqing 400065)

Abstract IP Quality of Service (QoS) guarantee is realized by managing queue and avoiding congestion in network. In this paper, ARED is a technology of avoiding congestion. When the average queue length is longer than the minimum threshold, ARED will deposal some packets in a nonzero prob congestion ability and intend to prevent the network from congestion. ARED overcomes the global synchronization, improves the performance of networks. This paper makes the Quantitative Analysis of ARED Character in Network, witch is important reference to future telecommunication router.

Keywords Congestion, ARED, Queue algorithms

1 引言

队列存在于路由交换设备中,其目的是在网络有足够的资源转发数据包出去之前保存数据包。只有在应用需求的带宽比网络能提供的带宽更多时,才会启动排队机制;反之,则会立即转发数据包。在网络队列中,数据包以任意的速率进入队列,以恒定的速率发送出去,这个速率不能超过接口的速率。队列只能保留预先定义的数量信息,当队列装填时,数据包按接收的顺序放进队列。当进入队列的数据包的数量超出了队列的容量时,路由器开始丢弃数据包。网络中的拥塞来源于网络资源和网络流量分布的不均衡性,为此需要配置合适的队列长度、队列算法和选择合理的丢弃机制,尽量减少网络拥塞,并提高网络性能和网络资源的利用率。本文下面主要对以随机早检测(RED)算法为基础得到的自适应 RED (ARED)算法进行探讨,这对路由器设计的算法选用和参数配置具有一定的参考价值。

2 拥塞回避

先进先出(FIFO)队列排队采用尾部丢弃,由于尾部丢弃会产生全局同步,导致网络吞吐量周期变化,即慢慢增加,到了拥塞点时被所有的会话拥塞,然后又回到原来的位置。这导致了对链路带宽的次优利用。拥塞回避技术试着减少这种全局同步,当链路达到拥塞状态时,它随机地从 TCP 会话中丢弃一些数据包,迫使 TCP 滑窗口减少发送窗口,降低发送流量。拥塞回避常采用随机早期检测(RED)和加权随机早期检测(WRED)技术,现在也有很多的路由器开始采用 ARED 算法。采用这些机制抑制了发送速率,整个链路的带宽利用率接近最优化的值。结果是更好地利用了网络资源和增加了整个期间的吞吐量。图1显示了与全局同步过程相比较的拥塞回避结果^[1]。

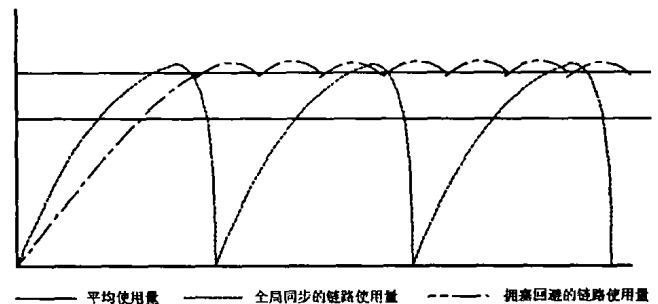


图1 拥塞回避与全局同步的对比

通过研究发现,拥塞回避技术有以下优点:防止在某些环境中发生拥塞;最大限度地利用了链路带宽;通过它的优先权提供优先等级。

拥塞回避技术的缺点是:只能使用于基于 TCP 的会话。其他的协议(如 IPX)不使用滑窗口的概念。当面临数据包丢弃的问题时,这些协议简单地以原来的速率重发数据包。而 RED 和 WRED 算法在大部分基于非 TCP 协议的网络中效率很低。

3 随机早期检测(RED)算法

RED 是由 Sally Floyd 和 Van Jacobson 提出的一种拥塞避免机制,它是一种积极的队列管理技术,旨在提供比传统的尾丢弃方式更佳的性能。它采用了预防拥塞的方法,并使用平均队列长度来决定是否丢弃分组,用分组丢弃几率来决定路由器丢弃分组的频度。在平均队列长度超过最小阈值后,开始以非零丢弃几率丢弃分组,而不是等到队列完全填满之后才开始丢弃分组。丢弃几率确保 RED 随机地丢弃少数几个流中的分组,避免了全局同步。RED 的主要目标是:

^{*}基金项目:重庆市科委应用基础项目;重庆市中青年骨干教师基金项目。刘晏兵 研究生,主要从事宽带网络性能分析及设计研究。

·拥塞避免: RED 的设计是为了避免拥塞,而不是对拥塞作出反应。这样, RED 必须检测拥塞的开始,以使网络维持在低时延和高吞吐量的工作区域;

·全局同步避免: 得知拥塞开始后, 路由器必须决定当通知哪个连接或哪些连接要退避。在当前的实现中, 通知是隐式的, 是通过丢弃分组来提供的。由于及早检测出拥塞, 并且通知必要的那些连接, 因此不会出现全局同步;

·没有偏见的支持突发通信: 它避免了只选择一些正在到达的分组来丢弃造成的对突发源站不利影响;

·对平均队列长度的限制: RED 应当能够控制平均长度, 因而能够控制平均时延。

4 自适应的 RED (ARED) 的定性讨论

4.1 原理

随机早检测 (RED) 机制的一个主要目标是使用队列平均长度和拥塞早通知来实现低时延和高带宽。仿真表明, 它在这方面很成功。但是 RED 存在两个很重要的缺陷。一个问题是 RED 对于参数的设置很敏感, 改变参数的设置会对性能产生很大的影响。到目前为止, 对 RED 中使用的参数还没有明确的设定方法。另一个问题是随着网络中“流”(Flow, 指一个 TCP 连接) 数目的增加, 网关的平均队列长度会逐渐增加。因此研究人员提出了自适应的 RED 算法。

自适应的随机早检测机制 (ARED) 算法^[2] 通过调节 Max_p 的值, 使平均队列长度保持在某一个目标长度 $target$ 附近。当平均队列长度大于目标长度时, 增大 Max_p 的值, 增加分组的丢弃概率, 使更多的分组不能入队, 反之, 减小 Max_p 的值, 使更多的分组入队。其原理如图 2 所示。

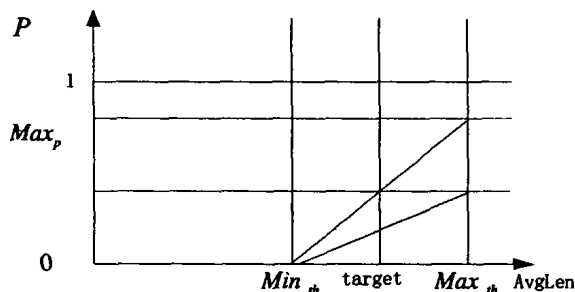


图2 ARED原理

ARED 具有 RED 的所有优点, 并且为队列提供可以预见的时延和吞吐量值并为各种流量类型提供合适的 QoS 服务级别。

4.2 ARED 适应算法描述

```
Every interval seconds:
If (AvgLen > target and Max_p <= 0.5)
Increase Max_p
Max_p = Max_p + alpha
Else if (AvgLen < target and Max_p >= 0.01)
decrease Max_p
Max_p = Max_p * beta
```

其中: $AvgLen$: 平均队列长度; $Interval$: 间隔时间; $Target$: RED 的目标队列长度; α : 增加因子, 为 $\min(0.01, Max_p/4)$; β : 减小因子, 一般为 0.9

4.3 参数讨论

(1) Max_p, Max_p 的上限 0.5 可以从两方面解释。首先当分组的丢弃概率超过 50% 后就会发生上述的类似“全局同步”的问题, 这并不利于我们解决问题。并且, 因为我们以一种缓和的方式在使用随机早检测机制, 我们希望当平均队列大小从 Min_{th} 变化到 Max_{th} 时, 分组丢弃概率从 0 变化到 Max_p ; 当平均队列大小从 Max_{th} 变化到两倍的 Max_{th} 时, 分组丢弃概率从 Max_p 变化到 1。因此, 将 Max_p 设置为 0.5, 当平均队列大小从

Min_{th} 变到两倍的 Max_{th} 时, 分组丢弃概率从 0 变到 1。 Max_p 的下限 0.01 是为了限制 Max_p 的范围而定的, 而且当分组的丢弃概率很小的时候, 用 0.01 来表示就已经足够了, 因为当平均队列大小在 $target$ 之下时, 情况是最好的。

(2) α 和 β 。注意到至少需要 $0.49/\alpha$ 个时间间隔, Max_p 才能从 0.01 增大到 0.5。同样的, 需要至少 $\log 0.02/\log \beta$ 个时间间隔, Max_p 才能从 0.5 降低到 0.01。对这两个参数的修改, 研究表明, Max_p 的一次改变不能使平均队列大小从 $target$ 外降到 $target$ 内, 或者从 $target$ 内上升到 $target$ 外。

5 拥塞控制的新思路

当前大多数的主动队列管理 (AQM) 技术集中于在路由器提供更高的吞吐量而不注重考虑队列时延。RED 和 ARED 都对进入流提供相同的待遇并且都企图获得高的吞吐量而不考虑正在传输的应用的时延要求。

新的思路是对 ARED 进行扩展, 不像 ARED 那样固定目标队列大小, 而是像下图那样基于进入业务的要求更新目标队列的大小。这样, 当进入业务大部分是吞吐量敏感业务时, 保持高的平均队列长度以提高整体的吞吐量。另一方面, 当进入业务大部分是时延敏感的业务时, 降低平均队列大小来减小队列的时延。当是中等吞吐量和中等时延敏感业务的情况下, 保持平均队列的大小使它能提供中等的吞吐量和时延性能, 即 $Target$ 是活动的, 如图 3 所示。

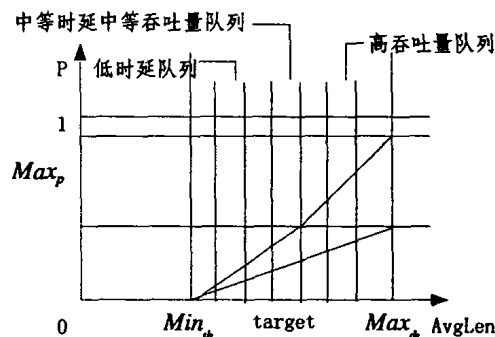


图3 新的思路原理

小结 本论文重点研究了拥塞避免以及分组丢弃策略: RED、ARED。ARED 是一种积极的队列管理算法, 使得路由器能够在队列溢出之前发现拥塞。本文定性探讨了 ARED 的参数: 最小阈值, 最大阈值, 平均队列长度, 最大丢弃概率等。但它的不足是 QoS 服务级别仍然是和队列绑定在一起, 流只能根据分类标准来获得相应队列的 QoS 服务, 而不能根据流业务本身的 QoS 要求给予服务, 只能从已有的配置了的服务中进行选择; 而且 ARED 的低时延是以牺牲链路利用率为代价的。在此基础上本文给出一些新的思路。

参考文献

- 1 宁科, 等译. IP 网络的 Cisco QoS 管理. 机械工业出版社, 2002
- 2 Phirke V. Traffic Sensitive Active Queue Management for Improved Quality of Service. A thesis submitted to the faculty of the WORCESTER POLYTECHNIC INSTITUTE In partial fulfillment of the requirements for Degree of Master of Science in Computer Science. May 2002
- 3 Chen J S, Guerin R. Performance study of an input queueing packet switch with two priority classes. IEEE Trans. Commun [J], 1991, 39(1): 117~126
- 4 Hluchj M G, Karol M J. Queueing in high-performance packet-switching. IEEE J. Sel. Areas Communication [J], 1998, 6(9): 1587~1597
- 5 林闯, 等, 因特网区分服务及其几个热点问题的研究. 计算机学报, 2000, 56(4)
- 6 WWW. IEEE. ORG