

# XCP 中 $\alpha$ 参数的自适应动态调整算法

陈元琰, 吴东, 陶小梅

(广西师范大学数学与计算机科学学院, 桂林 541004)

**摘要:** 分析了 XCP 在  $\alpha$  参数设置上存在的问题, 提出了一种基于平均队列变化、动态调整参数  $\alpha$  的参数自适应算法。NS2 模拟结果表明, 该算法能提高 XCP 的带宽利用率并且保持稳定。

**关键词:** XCP; TCP/IP; 拥塞控制

## Adaptive Dynamic Adjustment Algorithm of Alpha Parameter in XCP

CHEN Yuanyan, WU Dong, TAO Xiaomei

(College of Mathematics and Computer Science, Guangxi Normal University, Guilin 541004)

**【Abstract】** This paper analyzes the problem about setup of parameter alpha in XCP and proposes a parameter adaptive algorithm, which dynamically adjusts parameter alpha based on average queue change. Simulation result by NS2 show that the new algorithm could enhance bandwidth utilization of XCP, and keeps stable.

**【Key words】** XCP; TCP/IP; Congestion control

Explicit Control Protocol是针对高带宽乘积网络而提出的一种新协议<sup>[1]</sup>。XCP协议扩展了ECN显式拥塞指示机制, 在路由器引入拥塞控制器和公平控制器, 将拥塞控制从带宽分配策略中分离出来, 在端节点数据包结构中增加了H\_cwnd、H\_rtt和H\_feedback 3个域, 分别用于表示拥塞窗口、RTT估计和拥塞反馈。发送端发送数据包时使用H\_feedback域请求希望得到的拥塞窗口调整的变化量, 数据包途经的路由器根据当时网络可用带宽的状况修改H\_feedback域的值, 使用MaxNet网络结构, 即当数据包到达接收端时, H\_feedback域保留的是该数据包在途径各路由器中, 允许发送端可以增加的带宽最小值或要求发送端需要减少的带宽最大值。接收端通过ACK包将H\_feedback域的值反馈给发送端, 发送端按照反馈调整随后的发送速率。这种端节点和中间节点协作的执行模式以及解耦的拥塞控制体系使XCP获得了很好的网络性能。试验表明, 使用XCP协议可以获得很高的链路利用率、极小的排队队长和近于零的丢包率。

在分析 XCP 参数对网络性能的影响时, 本文发现在用户数量变化大及链路带宽不相同的网络环境中参数  $\alpha$  限制了链路带宽的利用率, 提出一种基于平均队列变化的动态调整参数  $\alpha$  的参数自适应算法。

### 1 对 XCP 拥塞控制器中参数 $\alpha$ 的分析

XCP 路由器中的拥塞控制器使用下式计算当前可用带宽<sup>[1]</sup>。

$$\phi = \alpha \cdot S - \beta \cdot Q / d$$

其中,  $\phi$  为计算得到的可用带宽;  $s$  为当前空闲带宽;  $Q$  为当前队列;  $d$  为控制器计算可用带宽的控制周期。

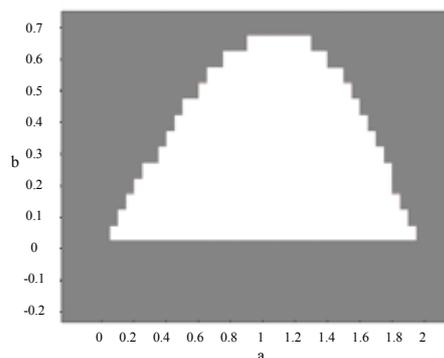
观察上式可知, 当前可用带宽是一部分空闲带宽减去清空当前队列所需带宽的剩余部分,  $\alpha$  为控制空闲带宽的参数,  $\beta$  则与清空当前队列所需的带宽相关。参数  $\alpha$  和参数  $\beta$  对网络的性能有着重要的影响。

文献[1]对 XCP 稳定性的分析指出, 当网络中数据流的

数量稳定、所有数据流均通过一条链路且 RTT 相同时, 只要参数  $\alpha$  与参数  $\beta$  满足以下条件, 网络的稳定性就与数据流数目、RTT 及链路带宽无关。

$$0 < \alpha < \pi / 4\sqrt{2} \text{ 且 } \beta = \alpha^2\sqrt{2}$$

文献[1]推导参数  $\alpha$  与参数  $\beta$  的取值范围时, 所作的假设在现实的网络环境下很难得到满足, 文献[2]给出了时延不同的网络环境中使网络稳定的  $\alpha$  与  $\beta$  的取值范围, 如图 1 所示, 白色区域为稳定区域, 黑色区域为不稳定区域。



XCP 在实际执行时参数  $\alpha$  和参数  $\beta$  的取值分别为 0.4 和 0.226。在不同网络环境中使用 XCP 传输数据网络, 信息最终都可以达到稳定, 但在一些用户数量变化大及链路带宽不相同的网络环境中出现了利用率不能进一步提高的现象。原因是, 在某个时间段内网络虽然处于稳定状态, 但保守的参数  $\alpha$  限制了最大使用空闲带宽的可能, 从而阻止了利用率的提高。在网络达到稳定的同时, 适当增大参数  $\alpha$ , 从空闲带

**基金项目:** 国家自然科学基金资助项目(70571017)

**作者简介:** 陈元琰(1961 -), 男, 副教授、博士, 主研方向: 计算机图形学, 计算机网络; 吴东、陶小梅, 硕士

**收稿日期:** 2005-11-22 **E-mail:** yychen@mailbox.gxnu.edu.cn

宽中取出可用带宽的比例变大, 可用带宽随之增加, 各数据流的吞吐量也会相应的增大, 提高网络的利用率。下面的实验证明了分析结果。

实验 1 使用文献[1]的网络拓扑结构, 在图 2 中, S是发送端, R是接收端, S和R一一对应; Bottleneck链路带宽为 45Mbps, 时延为 10ms; 链路带宽  $L_1=L_2=L_3=L_4=4.5Mbps$ ,  $L_5=45Mbps$ , 时延  $d_1=d_2=d_3=d_4=90ms$ ,  $d_5=10ms$ ; XCP协议参数  $\beta=0.226$ ,  $\gamma=0.1$ ; 模拟时间  $T=15s$ 。

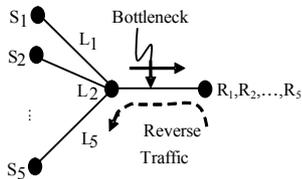


图 2 实验 1 的网络拓扑结构

(a)  $a=0.56$

(b)  $a=0.4$

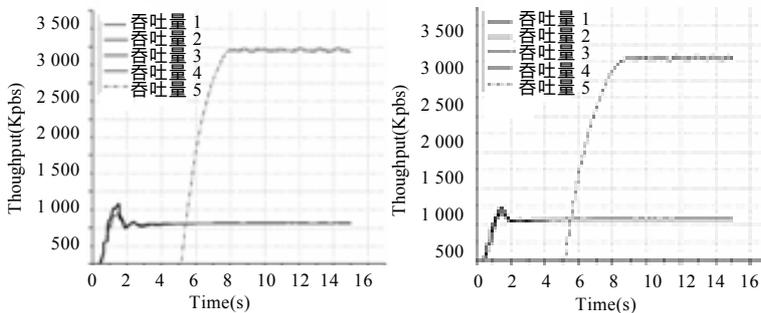


图 3 XCP 数据流的吞吐量

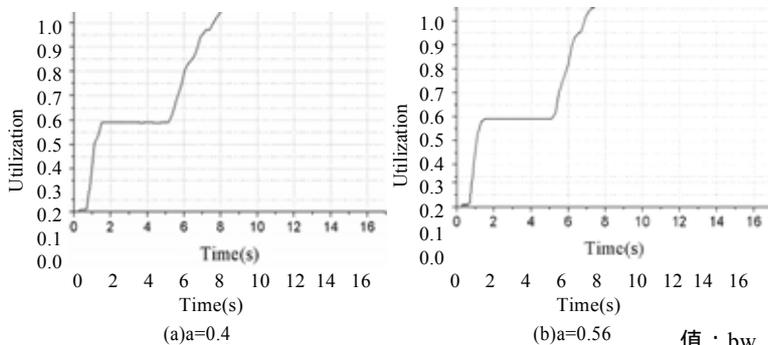


图 4 XCP 带宽利用率

$t=0$  时刻, 结点  $S_1 \sim$  结点  $S_4$  开始发送数据, 因为  $L_1=L_2=L_3=L_4=4.5Mbps$ , Bottleneck 链路带宽为 45Mbps, 所以 Bottleneck 链路还有空闲带宽。5s 后  $S_5$  开始发送数据, 由于  $L_5=45Mbps$ , 因此第 5 个数据流的吞吐量比其它数据流要大。当网络达到稳定状态时, 路由器中的队列很小, 拥塞控制器中的参数  $\beta$  产生的影响不大, 可以单独考察参数  $\alpha$  对网络稳定性及其带宽利用率的影响。参数  $\alpha$  分别为 0.56 和 0.4 时的实验结果如图 3 所示。

随着参数  $a$  的增大各数据流的吞吐量也都相应的增大, 但其仍保持了网络的稳定性。比较图 4 可以看出, 当  $a=0.4$ , 网络达到稳定时 Bottleneck 链路的带宽利用率为 87%, 当  $a=0.56$ , Bottleneck 链路的带宽利用率上升到 90%。

通过对图 3、图 4 的比较, 可以推断当网络达到稳定后, 适当地增大参数  $a$ , 可以在网络达到稳定的同时使各数据流的吞吐量增大, 提高链路的利用率。

## 2 参数自适应算法

根据上述对参数  $a$  在用户数量变化大和在链路带宽不相同的网络环境中限制利用率提高的问题所作的分析。本文提出了一种动态调整参数  $a$  的自适应算法(参数自适应算法), 网络在稳定的基础上, 提高主干链路的带宽利用率, 改善网络性能。该算法的主要思想是:

- (1) 当网络处于相对稳定的状态且链路尚有更多的空闲带宽时, 适当增大参数  $a$  的值, 以提高网络的带宽利用率;
- (2) 当网络抖动明显处于不稳定时, 适当减小参数  $a$  的值, 以恢复网络的稳定。

参数自适应算法要解决的首要问题是怎样判断当前网络处于相对稳定的状态。本文使用时延变化、数据到达速率、实际队列长度等指标判断网络是否稳定。因为时延变化的情况是端节点使用的判断条件, 计算数据到达速率会增加路由器的复杂度, 而实际队列长度对短期突发流量过于敏感, 不能准确反映网络状态, 所以通过检查路由器中平均队列长度的变化来判断网络是否稳定。平均队列长度是采用类似低通滤波器带权值的方法<sup>[4]</sup>计算的, 可以有效地将突发流引起的短期队列变化过滤。参数自适应算法描述如下:

```

Timer(free_alpha_time)
{
  Timer(free_queue_time)
  {
    calculate que_change;
    if(que_change < que_threshold)
    {
      calculate bw_surplus;
      if(bw_surplus > bw_threshold)
      {
        a=a+alpha_change;
      }
      else
      {
        a=a-alpha_change;
      }
    }
  }
}

```

其中  $free\_alpha\_time$  为调整参数  $a$  的周期, 是  $free\_queue\_time$  的整数倍, 可以防止频繁调整参数  $a$  造成网络不稳定;  $free\_queue\_time$  为计算平均队列长度的周期(时间等于 XCP 拥塞控制器中的控制周期), 多次计算并比较平均队列长度变化, 可以更准确地判断在  $free\_alpha\_time$  内网络是否稳定;  $que\_change$  为当前平均队列长度变化值;  $que\_threshold$  为平均队列长度变化阈值;  $bw\_surplus$  为当前空闲带宽;  $bw\_threshold$  为空闲带宽阈值;  $alpha\_change$  为参数  $a$  的调整值。

## 3 模拟实验

为了比较原算法和改进算法的性能, 本文使用 NS2 网络模拟软件进行了 2 次实验, 实验 2 反映了在用户数量稳定及链路带宽完全相同的网络环境下各算法的性能, 实验 3 反映了在用户数量不稳定及链路带宽不完全相同的网络环境下各算法的性能。在 2 次实验中, 参数自适应算法的参数设置如下:

$free\_alpha\_time = 20 * free\_queue\_time$ ;  $que\_threshold = 0.3$ ;  $bw\_threshold = 30\%$ ;  $alpha\_change = 0.1$ 。

比较图 3(a)和图 5(a)可以看出, 使用改进算法时各数据流的吞吐量都增大, 但依然保持了网络的稳定性。比较图 4(a)和图 5(b)可以看出, 网络达到稳定时利用率比改进前提高了约 6%。实验结果证明, 在用户数量不稳定及链路带宽不完全相同的网络环境中, 控制器对参数  $a$  进行自动调节的参数自适应算法使网络的带宽利用率得到了提高, 并且网络仍然保持良好的稳定性。

(下转第 153 页)