

# 一种增强 XCP 协议在多瓶颈网络拓扑环境下性能的方法

张慧翔 戴冠中 姚磊 潘文平

(西北工业大学自动化学院 西安 710072)

**摘要** XCP 协议具有很高的效率和良好的最大最小公平性,已成为基于显式反馈的拥塞控制协议的一种标准。进一步研究表明,XCP 协议在多瓶颈网络拓扑环境下,当下游存在瓶颈链路时,其性能变差,表现为带宽浪费、收敛时间延长以及公平性迅速恶化。深入分析了 XCP 协议在单瓶颈拓扑和多瓶颈拓扑环境下的收敛过程,指出了其性能变化的原因,进一步给出了一种结合相对反馈的改进方案,使端系统在收敛阻塞时,能根据瓶颈链路的网络相对负载更新其拥塞窗口。仿真实验表明,改进方案有效地提高了 XCP 协议在多瓶颈网络拓扑环境下的性能。

**关键词** 显式控制协议,拥塞控制,多瓶颈网络拓扑,显式相对反馈,动态分析

## Improving XCP Performance for Multi-bottleneck Topology

ZHANG Hui-xiang DAI Guan-zhong YAO Lei PAN Wen-ping

(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract** Owing to the high utilization and well max-min fairness, the eXplicit Control Protocol(XCP) has become one of the standard protocols based on explicit feedback. But further research shows that the performance of XCP degrades in the multi-bottleneck topology. For example, the bottleneck bandwidth is wasted and some flows can't achieve their max-min fair rate. The convergence behaviors of XCP flows in multi-bottleneck topology and the cause of such behaviors were analyzed first, and then a method with explicit relative feedback was proposed. When the convergence of the system is blocked, end system can update its congestion window according to the relative load of bottleneck link. Simulation shows the method effectively improves XCP performance for the multi-bottleneck topology.

**Keywords** eXplicit control protocol, Congestion control, Multi-bottleneck topology, Explicit relative feedback, Dynamical analysis

## 1 引言

随着高带宽延迟乘积网络的迅速发展,传统的基于隐式反馈的 TCP 协议表现出很多不足,采用显式反馈的拥塞控制协议得到迅速发展,出现了大量新的拥塞控制算法<sup>[1]</sup>,其中 XCP 协议<sup>[2]</sup>具有很高的效率和良好的最大最小公平性<sup>[9]</sup>,已成为基于显式反馈拥塞控制算法的一种标准。

XCP 引入了一个拥塞头部<sup>[3]</sup>。数据发送端在发送数据时填充拥塞头部中自身的状态信息(包括当前估计 RTT 值和拥塞窗口值),设置需求发送速率字段。在数据包传输路径上的路由根据网络状况更新需求发送速率。XCP 路由系统提供了两个控制器,效率控制器和公平控制器。效率控制器每隔一个控制周期  $d$  重新计算空闲带宽  $\phi$ ,有:

$$\phi = -\alpha(y - C) - \beta \frac{q}{d} \quad (1)$$

其中  $y$  表示每个控制周期的链路数据流量,  $q$  表示每个控制周期的持久队列长度,  $C$  为链路带宽,  $\alpha = 0.4$ ,  $\beta = 0.226$ 。为了提高新加入的数据流的收敛速度,在每个控制周期有  $h = \max(0, \gamma y - |\phi|)$  的 Shuffling Bandwidth<sup>[2]</sup> 也被重新分配,其中  $\gamma = 0.1$ 。

公平控制器根据加增乘减 AIMD(Additive Increase Multiplicative Decrease)<sup>[4]</sup> 的原则将空闲带宽分配到数据包。

XCP 路由算法将反馈分为正反馈和负反馈,分别对应 AI 和 MD 策略。当数据包流经所有的链路到达接收端,其拥塞包头中需求发送速率字段保存了传输路径上瓶颈链路的分配值。数据接收端拷贝包头中的需求发送速率字段到反馈速率字段,反馈回发送端。发送端根据反馈速率字段更新其拥塞窗口。XCP 协议是由路由系统直接控制端系统拥塞窗口的大小,这种分配方式带来了很高的链路利用率,并且队长基本为 0,没有丢包现象,同时具有良好的最大最小公平性。然而文献[5,6]进一步研究表明,XCP 协议在多瓶颈网络拓扑环境下达不到其原有的高效率和最大最小公平。

本文对该问题进行了深入分析,给出了一种显式相对反馈的改进算法,有效地弥补了 XCP 在多瓶颈网络拓扑环境下的不足。文章组织如下,第 2 节分析了 XCP 在单瓶颈和多瓶颈网络拓扑环境下的收敛行为以及 XCP 出现不足的原因。第 3 节给出了改进算法。第 4 节用仿真验证了改进算法的有效性。最后总结了我们的工作和下一步的目标。

## 2 XCP 协议收敛分析

### 2.1 单瓶颈网络拓扑下数据流的收敛分析

XCP 数据流加入到一个只存在单瓶颈的空闲网络中,瓶颈链路带宽为  $C$ ,有  $N$  个数据流通过该节点。XCP 路由算法按照 AIMD 的原则将正反馈量平均分配给所有数据流,将负

张慧翔 博士研究生,研究方向为 Internet 网的拥塞控制和 QoS 控制;戴冠中 教授,博士生导师,研究方向为自动控制、信息安全;姚磊 博士研究生,研究方向为网络分布式系统;潘文平 博士研究生,研究方向为 Web QoS 控制。

反馈量按照各数据流吞吐量的比例进行分配。用  $P_l, N_l$  表示链路  $l$  在一个控制周期内的正反馈量和负反馈量,有:

$$P_l = h + \max(\phi, 0) = \max(0, \gamma y - |\phi|) + \max(\phi, 0)$$

$$= \begin{cases} 0 & \phi \leq -\gamma y \\ \gamma y - |\phi| & -\gamma y < \phi \leq 0 \\ \gamma y & 0 < \phi \leq \gamma y \\ \phi & \gamma y < \phi \end{cases} \quad (2)$$

$$N_l = h + \max(-\phi, 0) = \max(0, \gamma y - |\phi|) + \max(-\phi, 0)$$

$$= \begin{cases} |\phi| & \phi \leq -\gamma y \\ \gamma y & -\gamma y < \phi \leq 0 \\ \gamma y - \phi & 0 < \phi \leq \gamma y \\ 0 & \gamma y < \phi \end{cases} \quad (3)$$

用  $x_n$  表示第  $n$  个控制周期时端系统发送速率,有:

$$x_n = x_{n-1} + \frac{P_l}{N} - x_{n-1} \frac{N_l}{y} \quad (4)$$

由公式(2)(3)(4),可以将数据流的收敛分为两个阶段。阶段一为效率收敛阶段,瓶颈链路处于  $\gamma y < \phi$  状态,有  $P_l = \phi, N_l = 0$ 。数据流从起始发送速率  $x_0$  增长,直到路由节点进入  $0 < \phi \leq \gamma y$  状态,即阶段二。阶段二为公平收敛阶段,路由节点处于  $0 < \phi \leq \gamma y$  状态,有  $P_l = \gamma y, N_l = \gamma y - \phi$ ,数据流发送速率继续增长直到其正负反馈量相等。在这种单瓶颈网络拓扑下,瓶颈利用率达到 100%,各个数据流的发送速率达到最大最小公平,为  $\frac{C}{N}$ 。

**定理 1** 在效率收敛阶段和公平收敛阶段的状态切换时刻,瓶颈链路的带宽利用率  $u = 0.8$ 。

证明:当两个阶段发生切换时,瓶颈链路有  $\phi = \gamma y$ ,利用公式(1),有  $\alpha(c-y) + \beta \frac{q}{d} = \gamma y$ 。在效率收敛阶段,数据流量小于路由处理能力,即  $y < c$ ,显然有队长  $q = 0$ ,因此有:  $u = \frac{y}{c} = \frac{\alpha}{\alpha + \gamma} = 0.8$ 。

**定理 2** 单瓶颈网络拓扑中,瓶颈节点利用率从 0 到完成效率收敛阶段耗费的控制周期数只与  $\alpha$  有关。

证明:在效率收敛阶段,数据流按照 AI 原则分配正反馈量,用  $x_n$  表示第  $n$  个控制周期时端系统发送速率,瓶颈链路队长  $q = 0$ ,由公式(4)有:

$$x_n = x_{n-1} + \frac{\alpha(C - N x_{n-1})}{N} = (1-\alpha)x_{n-1} + \frac{\alpha C}{N}, \text{迭代可解:}$$

$$x_n = (1-\alpha)^n (x_0 - \frac{C}{N}) + \frac{C}{N} \quad (5)$$

一般而言  $x_0 \leq \frac{C}{N}$ ,式(5)可近似为:

$$x_n = \frac{C}{N} (1 - (1-\alpha)^n)$$

因此瓶颈节点利用率为:  $u_n = \frac{N x_n}{C} = 1 - (1-\alpha)^n$ ,可解得:

$$n = \frac{\lg(1-u_n)}{\lg(1-\alpha)} \quad (6)$$

由定理 1,效率收敛阶段切换时刻  $u_n = 0.8$ ,再由公式(6)知,在单瓶颈网络拓扑中,瓶颈节点利用率从 0 到完成效率收敛阶段耗费的控制周期数只与  $\alpha$  有关。

XCP 在公平收敛阶段利用 AIMD 原则,使发送速率存在差异的数据流收敛到同一数值。在当前假设条件下,从效率

收敛阶段切换到公平收敛阶段时,各个数据流的发送速率相等,即  $x_n = \frac{0.8C}{N}$ ,由公式(4)知瓶颈节点分配给各个数据流的正负反馈量相等,不存在公平性收敛问题。各个数据流在公平收敛阶段收敛到公平速率  $\frac{C}{N}$ 。XCP 协议中  $\alpha$  默认为 0.4,可见 XCP 协议在单瓶颈网络拓扑环境下具有很快的收敛速度和很高的效率。

## 2.2 XCP 出现不足的多瓶颈网络拓扑环境

在多瓶颈网络拓扑环境下,对于某个链路  $L$ ,通过  $L$  的数据流可以归为 3 个集合:集合  $\varphi$  表示以  $L$  的下游链路为瓶颈的数据流,集合  $\chi$  表示以  $L$  的上游链路为瓶颈的数据流,集合  $\psi$  表示以  $L$  为瓶颈的数据流。如果所有数据流以下游链路和链路  $L$  为瓶颈,则系统平衡时,链路  $L$  分配给集合  $\varphi$  中数据流的带宽要大于下游瓶颈链路分配的带宽,否则所有数据流都以链路  $L$  为瓶颈。当数据流到达下游瓶颈,链路  $L$  分配给集合  $\varphi$  中数据流的带宽被缩减为下游瓶颈链路分配的较小值,而链路  $L$  无法探知下游存在瓶颈,直接导致链路  $L$  的带宽利用率下降。文献[5]进一步证明,在现有 XCP 协议参数设置条件下,链路  $L$  的利用率只能达到 80%,并且随着集合  $\varphi$  中数据流数量的增加,集合  $\psi$  中数据流享有的带宽会远低于理论的最大最小公平分配值。

本文对这种下游存在瓶颈链路的多瓶颈拓扑(Downstream case<sup>[5]</sup>)进行分析。图 1 描述了一种基本的 Downstream 拓扑结构,定义如下:

- 1) Link0 为当前瓶颈链路,Link1 为下游瓶颈链路;
- 2)  $N = N_i + N_j$  个数据流流经 Link0,其中  $N_i$  个数据流流经 Link1,剩余的  $N_j$  个数据流从节点 R1 分流;
- 3) 流经 Link1 的  $N_i$  个数据流以 Link1 为瓶颈链路,剩余的  $N_j$  个数据流以 Link0 为瓶颈链路;
- 4)  $\frac{C_0}{N} > \frac{C_1}{N_i}$ ,以保证 Link1 先于 Link0 收敛到平衡。

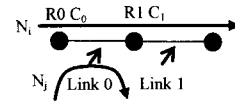


图 1 下游存在瓶颈链路的多瓶颈拓扑

## 2.3 Downstream case 环境下 XCP 收敛分析

将流经一条链路的数据流分为两种,以该链路为瓶颈的数据流称为瓶颈流,以其它链路为瓶颈的数据流为非瓶颈流。对 Link0,数据流  $N_i$  为非瓶颈流, $N_j$  为其瓶颈流。假设图 1 描述的网络拓扑中所有链路的传输延迟相等,分析  $N_i + N_j$  个数据流同时加入到图 1 描述的空闲网络中数据流的收敛过程,分为 3 个阶段。

阶段一为 Link1 收敛阶段,流经 Link1 的  $N_i$  个数据流按照 2.1 节描述的单瓶颈收敛过程收敛至平衡。流经 Link1 的  $N_i$  个数据流以 Link1 为瓶颈,其发送速率稳定在其最大最小公平分配的数值,即  $x_i = \frac{C_1}{N_i}$ 。

对于 Link0 而言,在 Link1 收敛的过程中,由公式(2)(3)(4),其瓶颈流发送速率遵循下面的公式:

$$x_n^j = x_{n-1}^j + \frac{\alpha(C_0 - N_j x_{n-1}^j - N_i x_{n-1}^i)}{N} \quad (7)$$

$$\text{经整理有: } x_n^j = (1 - \frac{\alpha N_j}{N}) x_{n-1}^j + \frac{\alpha C_0}{N} - \frac{\alpha N_i}{N} x_{n-1}^i,$$

迭代可解得:  $x_n^i = A^n x_b^i + \frac{C_0}{N_j} - \frac{C_0}{N_j} A^n - \frac{\alpha N_i}{N} \sum_{k=0}^{n-1} A^{n-1-k} x_k^i$ ,  
其中  $A = 1 - \frac{\alpha N_i}{N}$ , 很明显有:  $x_n^i < \frac{C_0}{N_j} - \frac{C_0}{N_j} (1 - \frac{\alpha N_i}{N})^n$ .

因此可得利用率  $u_b^i$  为:

$$u_b^i = \frac{N_j x_n^i}{C_0} < 1 - (1 - \frac{\alpha N_i}{N})^n \quad (8)$$

从(8)式可见, 当 Link1 收敛至平衡时, 如果  $N \geq N_j$ , 则  $u_b^i$  趋近于 0, 表明 Link0 的瓶颈流占用其带宽比例很小。Link0 的带宽利用率满足:

$$u_0 = u_b^i + u_b^0 < 1 - (1 - \frac{\alpha N_i}{N})^n + \frac{C_1}{C_0}$$

Link0 带宽主要被其非瓶颈流占据, 如果  $N \geq N_j$ , 且  $\frac{C_1}{C_0} < 0.8$ , 可以保证当 Link1 收敛至平衡时, Link0 处于效率收敛阶段, 即  $u_0 < 0.8$ 。

阶段二, Link1 已经平衡, 其瓶颈流平衡速率为:  $x_n^i = \frac{C_1}{N_i}$ , 而 Link0 的瓶颈流继续收敛。由公式(2)(3)(4)有:

$$x_n^i = x_{n-1}^i + \frac{\alpha(C_0 - N_j x_{n-1}^i - N_i x_n^i)}{N}, \text{ 进一步有:}$$

$$x_n^i = (1 - \frac{\alpha N_i}{N}) x_{n-1}^i + \frac{\alpha}{N} (C_0 - C_1), \text{ 迭代可解得:}$$

$$x_n^i = A^n x_b^i + \frac{C_0 - C_1}{N_j} (1 - A^n) \quad (9)$$

其中,  $x_b^i$  表示阶段一结束时, Link0 瓶颈流的发送速率。由定理 1, 效率收敛结束时 Link0 的带宽利用率达到  $u_0 = 0.8$ , Link0 瓶颈流的发送速率收敛到:

$$x_n^i = \frac{0.8C_0 - C_1}{N_j} \quad (10)$$

由(9)(10)可解得阶段二收敛所需的控制周期数:

$$n = (\lg \frac{0.2C_0}{C_0 - C_1 - N_j x_b^i}) / \lg(1 - \frac{\alpha N_i}{N}) \quad (11)$$

由定理 2 知在单瓶颈网络中 XCP 协议的效率收敛速度取决于  $\alpha$ ; 而在 Downstream 拓扑环境下, 由公式(9)可见效率收敛速度取决于  $\frac{\alpha N_i}{N}$ , 如果  $N \geq N_j$ , 则需要很长的收敛时间。

如果取  $C_0 = 155\text{Mbps}$ ,  $C_1 = 100\text{Mbps}$ ,  $N_j = 1$ ,  $N_i = 64$ , 由公式(7)可得  $x_b^i = 4.11$ , 由公式(10)知  $x_n^i = 24$ , 由公式(11)知  $n = 80.3$ , 与 Link1 相比 Link0 经历很长一段时间完成效率收敛。在相同带宽条件下如果  $N_j = 1$ , 由公式(11), 随着 Link0 非瓶颈流  $N_i$  增加, 其效率收敛消耗的控制周期近似呈线性增长。图 2 的上图显示了这种趋势。

阶段二结束,  $u_0 = 0.8$ , Link0 瓶颈流继续收敛进入阶段三, 为 Link0 公平收敛过程。根据公式(2)(3)(4) Link0 的瓶颈流收敛遵循下面的规律:

$$x_n^i = x_{n-1}^i + \frac{\gamma y}{N} - \frac{\gamma y - \alpha(C_0 - y)}{y} x_{n-1}^i \quad (12)$$

在阶段三 Link0 瓶颈流收敛至平衡, 平衡时刻有  $x_n^i = x_{n-1}^i = x$ , 代入公式(12), 有:

$$\frac{\gamma y}{N} = \frac{\gamma y - \alpha(C_0 - y)}{y} x \quad (13)$$

其中  $y = C_1 + N_j x$ , 进一步整理得:

$$Ax^2 + Bx + C = 0, \text{ 其中 } A = (\alpha + \gamma)NN_j - \gamma N_j^2 \\ B = \gamma NC_1 - \alpha N(C_0 - C_1) - 2C_1 N_j \gamma, C = -\gamma C_1^2, \text{ 解方程可得:}$$

$$x = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (14)$$

如果取  $C_0 = 155\text{Mbps}$ ,  $C_1 = 100\text{Mbps}$ ,  $N_j = 1$ ,  $N_i = 64$ , 由式(14)得:  $x \approx 26$ , 而理论最大最小公平分配  $x = \frac{C_0 - C_1}{N_j} = 55$ 。

可见在阶段三 Link0 瓶颈流速率增长很小, Link0 空闲带宽大部分被浪费, Link0 无法收敛到最大最小公平。在相同带宽条件下如果  $N_j = 1$ , 由公式(14), 随着 Link0 非瓶颈流  $N_i$  增加, Link0 瓶颈流享有的带宽迅速减少, 出现严重的不公平现象, 图 2 的下图显示了这种趋势, 结果与文献[5]一致。

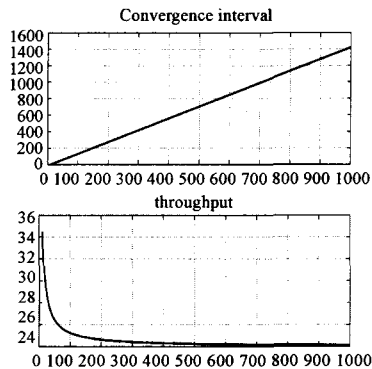


图 2 收敛速度和系统平衡时速率的理论值

总而言之, Downstream 拓扑带来了 3 个问题:

- 1) 当前链路带宽的浪费, 其瓶颈流收敛阻塞;
- 2) 当前链路瓶颈流收敛时间变长;
- 3) 当前链路瓶颈流无法收敛到其最大最小公平带宽。

造成 Downstream 拓扑下 XCP 性能降低的原因在于: XCP 是一种绝对数值反馈, 反馈的数值直接控制着端系统拥塞窗口的大小。当前链路无法感知下游存在瓶颈, 但流经当前链路的非瓶颈流速率受其下游瓶颈节点所控制, 当前链路所分配的带宽被下游瓶颈链路覆盖。在 Downstream 拓扑中, 当系统平衡时, Link0 分配给瓶颈流反馈的数值为 0, 但实际上有 20% 左右的带宽没有被利用。因此如果能够将 Link0 中 20% 的带宽没有被利用的信息反馈给端系统, 则端系统就能提高拥塞窗口大小, 提高链路利用率和公平性。

### 3 结合相对反馈的改进方法

在基于显式反馈的拥塞控制算法中, 采用相对反馈的算法有很多, 如 VCP<sup>[7]</sup>, MaxNet<sup>[8]</sup>。这些算法都是在路由系统计算一个反映当前链路负载的指标, 端系统则根据该指标调节拥塞窗口。本文借鉴了这种相对反馈的思想, 使端系统能根据瓶颈链路的负载状况更新其拥塞窗口。

首先系统需要区分 XCP 正常收敛和 Downstream 拓扑收敛这两种情况。系统初始默认为 XCP 正常收敛状态。在 XCP 的每个控制周期, 计算一个反映当前链路负载状况的相对指标负载因子  $\rho$ , 公式如下:

$$\rho = \frac{y + 0.5q}{Cd} \quad (15)$$

其中,  $y$  为上一个控制周期内链路入口总量,  $C$  为链路带宽,  $d$  为控制周期,  $q$  为平均队列长度, 采用指数加权平均方式更新该队长。利用前后两个周期的负载因子值, 计算其差值的绝对值  $\Delta\rho$ , 采用指数加权平均得到平滑后的值  $\theta$ , 即:  $\theta = (1 - \kappa)\theta + \kappa\Delta\rho$ , 其中  $\kappa$  取 0.3。

如果  $\theta < 0.01$  表示负载因子变化很小, 意味着当前链路收敛缓慢或者阻塞。这可以分为三种情况: 一是 XCP 正常收敛达到平衡, 此时负载因子  $\rho$  会维持在 1.0 左右; 二是 Down-

stream 拓扑收敛阻塞,假设  $\rho$  阻塞到  $\sigma$ ,根据网络参数设置  $\sigma \in (0.8, 1.0)$ ;三是 Downstream 拓扑缓慢收敛,也就是链路负载因子缓慢收敛到  $\sigma$  的过程,有  $0 < \rho < \sigma$ 。为了区分 XCP 正常收敛和 Downstream 拓扑收敛阻塞两种情况,设置 0.96 为  $\sigma$  的上界。如果 Downstream 拓扑收敛阻塞在 0.96 和 1.0 之间,系统按照 XCP 正常收敛处理。因此根据阻塞时的负载因子  $\rho$  分为四种状态:

- 1) 低利用率状态,编码为‘01’, $0 < \rho < 0.8$ ,表明链路存在较多的空闲带宽,端系统可较快地增大拥塞窗口。
- 2) 高利用率状态,编码为‘02’, $0.8 \leq \rho \leq 1.0$ ,表明链路存在较少的空闲带宽,端系统可较慢地增大拥塞窗口。
- 3) 饱和状态,编码为‘03’, $\rho > 1.0$ ,表明链路已经饱和,端系统需要减少拥塞窗口。
- 4) 如果链路前一状态为 XCP 正常收敛状态,且  $0.96 \leq \rho \leq 1.0$ ,表明链路为 XCP 正常收敛至平衡,编码为‘00’。

如果  $\theta > 0.01$ ,表明当前链路处于收敛过程之中。如果链路前一状态为 XCP 正常收敛,则链路状态保持不变;否则根据  $\rho$  更新链路状态,更新原则与上述的前三个条件一致。

系统利用 XCP 拥塞头部<sup>[3]</sup>中保留的 8 个比特位的两位保存链路状态编码。每个数据包离开当前节点时,首先比较 XCP 反馈的绝对反馈量,如果当前链路 XCP 反馈量小于拥塞头部中携带值,则更新拥塞头部中 XCP 反馈量为当前链路值,同时更新拥塞头部中的链路状态编码为当前链路编码。如果当前链路 XCP 反馈量大于拥塞头部中携带值,则比较链路状态编码,如果当前链路状态编码大于拥塞头部携带值,则更新拥塞头部中链路状态编码为当前链路编码。

数据接收端收到 XCP 数据包,利用拥塞头部<sup>[3]</sup>中另外两个比特位将瓶颈链路系统状态编码反馈回发送端。发送端借鉴 VCP<sup>[7]</sup>的端系统算法,低利用率状态采用 MI(Multiplicative Increase)较快地增大拥塞窗口;高利用率状态,采用 AI(Additive Increase)较缓地增大拥塞窗口;饱和期状态,采用 MD(Multiplicative Decrease)减小拥塞窗口。如果处于 XCP 正常收敛状态,则直接采用 XCP 反馈值。相应算法描述如下:

$$MI: cwnd(t+rtt) = cwnd(t) \times (1+\epsilon)$$

$$AI: cwnd(t+rtt) = cwnd(t) + \alpha$$

$$MD: cwnd(t+\delta t) = cwnd(t) \times \beta$$

其中  $\epsilon = 0.0625, \alpha = 1, \beta = 0.85$ 。

这样端系统实际上获取了两个数值增量,一个为原有 XCP 协议反馈数值,一个为根据瓶颈链路状态编码计算的窗口增量。选取两者中较大值作为新的窗口增量。

## 4 仿真实验

利用网络仿真软件 NS2<sup>[10]</sup>,采用图 1 描述的拓扑,网络参数设置如下: $C_0 = 155\text{Mbps}, C_1 = 100\text{Mbps}$ ,所有链路传播延迟相等,为 50ms,所有数据流在同一时刻进入系统。

仿真实验 1:固定  $N_i = 64, N_j$  从 1 到 64 变化。比较了采用 XCP 协议和协议改进后 Link0 和 Link1 的平衡时的利用率,仿真结果见图 3。XCP\_R 代表采用改进后协议的曲线,XCP 代表 XCP 原有协议的曲线,横坐标为  $N_j$ 。

观察图 3,对于 Link0 而言,采用原 XCP 协议,其利用率随着  $N_j$  的减小从 1.0 减少到 0.81 左右;采用改进后的协议,Link0 的利用率能维持在 0.98 左右,表明改进后的协议有效地提高了多瓶颈拓扑环境下系统的利用率。对于 Link1 而言,当  $N_j = 64$  时,只有 Link0 为系统瓶颈,相当于单瓶颈拓

扑,Link1 收敛于 0.77 左右;当  $N_j \leq 32$  时,Link1 的利用率始终保持在 1.0 左右,说明改进后的协议不影响 XCP 正常收敛时的高利用率。

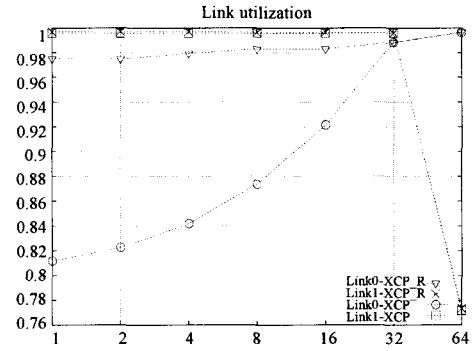


图 3 利用率曲线

仿真实验 2:固定  $N_i = 1, N_j$  从 1 到 128 变化。比较了采用 XCP 协议和协议改进后  $flow_i$  和  $flow_j$  平衡时的平均吞吐量与其对应最大最小公平分配值的比值,横坐标为  $N_i$ 。仿真结果见图 4。

观察图 4,对于  $flow_i$  而言,其比值保持在 0.95 左右,基本上都收敛到最大最小公平。对  $flow_j$  而言,采用原 XCP 协议其比值随着  $N_i$  的增大从 1 减小到 0.43 左右,表明出现了严重的不公平现象;采用改进后的协议,其比值能维持在 0.9 左右,有效地改善了系统的公平性。但由于改进协议采用基于端系统的 AIMD 算法,速率无法稳定在最大最小公平分配水平,而是在该值附近小幅振荡,使得该比值无法进一步提高。

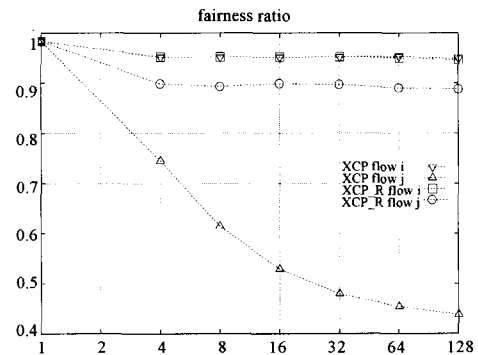


图 4 公平性曲线

**结束语** XCP 协议具有优异的性能表现,但如果当前链路无法感知下游存在瓶颈链路时会导致带宽的浪费,系统达不到最大最小公平。本文深入分析 XCP 协议在单瓶颈和多瓶颈网络拓扑结构下的收敛过程,结合相对反馈对 XCP 协议进行了改进,仿真实验证明了改进的有效性。为了使相对反馈和 XCP 绝对反馈模式更好地结合,需要对相对反馈的参数设置进行进一步的研究,这也是我们下一步的目标。

## 参考文献

- [1] 张慧翔,戴冠中,等.具有显示反馈的拥塞控制系统研究进展.计算机科学,2007,34(12):18-24
- [2] Katabi D, Handley M, Rohrs C. Congestion control for high bandwidth-delay product networks. ACM SIGCOMM Computer Communications Review,2002,32(4):89-102

(下转第 97 页)

得攻击协议、攻击方式和攻击强度等信息。

## 5 与相关工作比较

为了分析使用 HSMC-SVM 识别 DDoS 攻击的性能,此处将此算法与已有的工作进行比较,比较的项目为能否识别不同协议发起的攻击,能否指示攻击强度,能否指示攻击协议,能否指示攻击方式 4 个方面。

2003 年,Cheng Jin 提出基于网络包的 TTL 值的检测算法<sup>[11]</sup>;2004 年出现了基于网络相似度<sup>[12]</sup>、基于网络的自相似性<sup>[13]</sup>、基于网络包的 VDR 值<sup>[14]</sup>的检测算法;2005 年,出现了基于流连接密度 (FCD)<sup>[6]</sup>、基于神经网络<sup>[15]</sup>、基于多个 SVM<sup>[7]</sup>以及基于 SYN 包流量分布<sup>[16]</sup>的检测算法;2007 则出现了以 OWCD 来检测 DDoS 的算法<sup>[2]</sup>。比较的结果显示在表 3 中,“√”表示具有该功能,反之为“×”。

表 3 的结果显示,目前许多检测算法均将注意力集中到如何检测到攻击,而忽略了提供更多攻击信息的工作,因此显得检测工作并不全面。相比之下,使用 HSMC-SVM 的方法能更深入地检测 DDoS 攻击。

不过,用 HSMC-SVM 检测 DDoS 攻击,也有其不足之处。当出现不同的攻击行为,这种方法将失效。例如,我们曾将 MIX Flood 攻击中三种协议的比例改变后,再送入 HSMC-SVM 检测,则完全不能获得正确的检测结果,这是因为训练样本中的 MIX Flood 攻击的三种协议比例仅为 1:1:1。因此,如何提高 HSMC-SVM 检测方法的灵活性,仍是一个待研究的课题。

表 3 与相关工作的比较结果

Algorithm	Distinguish	Indicate	Indicate	Indicate
	DDoS	Density	Protocol	Pattern
FCD <sup>[6]</sup>	√	×	×	×
TRA <sup>[7]</sup>	√	×	×	×
ANN <sup>[15]</sup>	only UDP Flood	×	×	×
SYN Packets <sup>[16]</sup>	only SYN Flood	√	×	×
Likeness <sup>[12]</sup>	√	√	×	×
TTL <sup>[11]</sup>	√	√	×	×
Self-Similarity <sup>[13]</sup>	√	√	×	×
OWCD <sup>[2]</sup>	√	√	×	×
VDR <sup>[14]</sup>	√	√	√	×
HSMC-SVM	√	√	√	√

**结束语** 要在低虚警率和低误警率的条件下,有效防御 DDoS 攻击,需要检测环节能提供更多的攻击信息,而目前大多数的检测算法仅将注意力集中到如何检测到攻击,很难同时识别攻击强度、攻击协议和攻击方式等信息。本文通过多分类的方式,将攻击分为 24 种不同的类型,并利用快速训练算法 HSMC-SVM,实现了这一目标。实验表明,这种方法通过提供识别的类标,同时提供攻击强度、攻击协议和攻击方式等

较全面的攻击信息,为 DDoS 的防御环节针对不同类型的攻击进行有效防御提供了保障。

## 参考文献

- [1] Douligeris C, Mitrokotsa A. DDoS Attacks and Defense Mechanisms; A Classification // Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology. 2003; 190-193
- [2] Xu Tu, He Da-ke, Zheng Yu. Detecting DDoS Attack Based on One-way Connection Density // Proceedings of Tenth IEEE International Conference on Communication Systems. 2006
- [3] Cheng C-M, Kung H, Tan K S. Use of spectral analysis in defense against Dos attack // Proceedings of IEEE GLOBECOM. Division of Engineering and Applied Science Harvard University
- [4] Feinstein L, Schnachenberg D, Balupari R, et al. Statistical Approaches to DDoS Attack Detection and Response // Proceedings of the DARPA Information Survivability Conference and Exposition. 2003
- [5] Jin Shuyuan, Yeung D S. A Covariance Analysis Model for DDoS Attack Detection. IEEE Communications Society, 2004; 1882-1886
- [6] 孙钦东, 张德运, 高鹏. 基于时间序列分析的分布式拒绝服务攻击检测[J]. 计算机学报, 2005, 28(5): 767-773
- [7] Seo J, Lee C, Shon T. A New DDoS Detection Model Using Multiple SVM and TRA // EUC Workshops 2005. LNCS 3832. 2005; 976-985
- [8] 朱美琳, 刘向东, 陈世福. 用球结构解决多分类问题[J]. 南京大学学报: 自然科学版, 2003, 39(2): 153-158
- [9] Xu Tu, He Dake, Luo Yu. A New Orientation for Multi-Class SVM // Proceedings of the SNPD. 2007; 899-904
- [10] 徐图, 罗瑜, 何大可. HSMC-SVM 的二次逼近快速训练算法. 电子与信息学报(已录用)
- [11] Jin Cheng, Wang Haining, Shin K G. Hop-count Filtering: An Effective Defense Against Spoofed DDoS Traffic // Proceedings of the 10th ACM Conference on Computer and Communications Security. 2003
- [12] 何慧, 张宏莉, 张伟哲, 等. 一种基于相似度的 DDoS 攻击检测方法[J]. 通信学报, 2004, 25(7): 176-184
- [13] Xiang Y, Lin Y, Lei W L, et al. Detecting DDoS Attack Based on Network Self-Similarity // IEE Proc. Commun. 2004, 151(3): 292-295
- [14] Limwivatkul L, Rungsawang A. Distributed Denial of Service Detection Using TCP/IP Header and Traffic Measurement Analysis // Proceedings of International Symposium on Communications and Information Technologies. 2004; 605-610
- [15] Siaterlis C. Detecting incoming and outgoing DDoS attack at the edge using a single set of network characteristics // Proceedings of the 10th IEEE Symposium on Computers and Communication. 2005
- [16] Ohsita Y C, Ata YuiChi, Murata M. Detecting Distributed Denial-of-Service Attacks by Analyzing TCP SYN Packets Statistically // Global Telecommunications Conference. 2004; 2043-2049
- [6] Wang P, Mills D L. Simple analysis of XCP equilibrium performance // Proc. CISS 2006. Princeton, NJ, USA, 2006
- [7] Xia Y, Subramanian L, et al. One more bit is enough. ACM SIGCOMM Computer Communication Review, 2005, 35(4): 37-48
- [8] Wyrowski B P, Zukerman M. MaxNet: A congestion control architecture for max-min fairness. IEEE Commun. Lett., 2003, 6(11): 588-599
- [9] Welzl M. Network Congestion Control. John Wiley & Sons Ltd, 2005
- [10] The network simulator ns-2. 30. <http://www.isi.edu/nsnam/ns>

(上接第 72 页)

- [3] Falk A, Katabi D, Pryadkin Y. Specification for the Explicit Control Protocol (XCP). draft-falk-xcp-02. txt (work in progress), November 2006
- [4] Chiu D, Jain R. Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks. Journal of Computer Networks and ISDN, 1989, 17(1): 1-14
- [5] Low S, Andrew L, Wyrowski B. Understanding XCP: Equilibrium and fairness // Proc. IEEE INFOCOM. Miami, FL, USA, 2005