

基于跳数的公平性增强 RED 及其分类器实现

武航星¹ 郑雪峰¹ 潘文平²

(北京科技大学计算机与通信工程学院 北京 100083)¹ (南京航空航天大学自动化学院 南京 210016)²

摘要 为了同时兼顾改善不同往返时延(RTT)和多拥塞链路所引发的不公平性问题,在实际网络环境测量结果的基础上,提出了利用数据包所经过的路由器跳数来增强公平性的思路,并构造了一个二维两类分类器来对 RED 算法进行简单的修改,实现了一个公平性增强的 RED 算法。NS2 下的模拟表明,该算法可有效增强公平性,而且具有实现简单、易于在实际网络中部署的优点。

关键词 RED,公平性,分类器

中图法分类号 TP393 **文献标识码** A

Hop-counts-based Fairness Enhancement RED and its Classifier Implementation

WU Hang-xing¹ ZHENG Xue-feng¹ PAN Wen-ping²

(School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China)¹

(College of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)²

Abstract To solve unfairness bandwidth allocation problem induced by different RTT and multiple congestion links, we proposed a fairness enhancement RED(FERED) based on the measurement results in the real Internet in this paper. FERED utilizes the hop-counts to improve fairness and constructs a two-category classifier to modify the RED algorithm. Simulation results in NS2 show that FERED can efficiently improve fairness. Moreover, FERED can be deployed easily.

Keywords RED, Fairness, Classifier

1 引言

在过去几十年中, TCP 协议在网络的稳定运行方面起到了至关重要的作用。然而一些研究表明,在某些情况下, TCP 协议会导致瓶颈链路带宽在竞争数据流之间的分配存在严重的不公平性问题。Chiu 分析了具有不同往返时延(RTT)的 TCP 数据流之间的带宽分配不公平性^[1]; Floyd 的研究工作表明,在存在多拥塞链路的环境中,经过多个拥塞链路的 TCP 数据流与经过较少拥塞链路的 TCP 数据流相比,在瓶颈带宽竞争时处于明显的劣势地位,甚至可能被“饿死”^[2]。此后针对 TCP 存在的不公平性问题的研究一直是网络研究领域中的一个热点问题。研究者已经提出了很多的改善方法,例如“常速率”拥塞窗口(CWND)增加算法,将 CWND 在一个 RTT 内的增加因子改变为 $\alpha \times RTT$, 以对具有较大 RTT 的 TCP 数据流进行一定程度的保护,从而改善因 TCP 数据流具有不同 RTT 所引发的公平性问题^[2]。FAST TCP^[3]利用排队时延作为反馈信息在具有不同 RTT(Round-Trip Time)的数据流间实现了加权成比例公平性。文献[4]中提出的 FAST TCP 改进算法可实现更好的公平性,但该算法需要路由器提供显式的拥塞指示。PFWRR^[5]根据缓存的占用情况,在缓存足够时可在不同队列分组间实现排队时延的比例公平性,在缓存不足时可在不同队列分组间实现丢包

的比例公平性。在文献[6]中,作者采用与 PFWRR 类似的方法,在缓存足够和不足时,分别采用排队时延和丢包率作为拥塞信息,在不同数据流间实现了比例公平性。然而,正如上述所举实例,目前已有的众多算法大都针对不同 RTT 引发的不公平性问题进行改善,能够同时兼顾多拥塞链路引发的不公平性问题的算法还很少。随着网络负载的急剧增加,多拥塞链路环境下的不公平性问题将更易出现,不可忽视。在 TCC-EFE^[7]中,作者就验证了 RED, AVQ, REM, PI 等著名 AQM 算法在多拥塞链路环境下存在严重的不公平性问题,并提出了一个新颖的、利用模式识别中分类器思想实现 AQM 机制的方法。模拟试验显示其可极大地提高公平性,但是该机制的实现需要对 TCP 协议进行修改,在数据包 TCP 报头增加一个字段来表示 CWND,因此难以在网络中部署,缺乏可行性。

为了同时兼顾改善由不同 RTT 和多拥塞链路所引发的不公平性问题,本文在实际网络环境中测试实验结果的基础上,提出了利用数据包所经过的路由器跳数来增强公平性的思路。这里跳数指数据包已经经过的路由器数,该值可通过数据包自身 IP 报头中的 TTL 字段计算得到。通过沿用文献[7]中分类器实现 AQM 的方法,利用队列长度和一个与跳数相关的值 h_i 作为二维观测量,尝试构造了一个二维两类的分类器对 RED 算法进行简单的改进,实现了一个公平性增强的

到稿日期:2010-12-14 返修日期:2011-03-03

武航星(1978-), 博士后, 主要研究方向为网络拥塞控制和流量控制, E-mail: whx9711@163.com; 郑雪峰(1951-), 男, 教授, 主要研究方向为计算机网络、信息安全; 潘文平(1980-), 博士后, 主要研究方向为网络服务质量。

RED 机制。为方便引用,称之为 FERED(Fairness Enhancement RED)。NS2 下的模拟表明,FERED 可有效增强公平性,同时 FERED 的一个很好的优点是实现简单,无需对原有 TCP/IP 协议进行任何修改,易于在实际网络中部署。

2 用分类器实现 AQM 的思路

RED 算法根据平均队列长度这一个指标计算出数据包丢弃概率,并以此概率丢弃数据包,队长越长则丢弃概率越大。这的确是一个有效的方法,此后类似的方法也经常被其他 AQM 机制采用。但实际上,决定一个数据包被丢弃还是接收更应该看作是一个基于一定的目标进行决策的过程,在决策中依靠的信息越多,决策就越精确。例如除队列长度外,链路利用率、数据包到达速率等指标亦有助于做出更加准确的丢弃决策。然而困难之处在于这些不同的指标各自具有不同的属性,难以将它们用简单函数结合起来。正因为此,AQM 算法大都采用单一指标做出丢包决定。然而,在模式识别中的分类器思想却可以容易地结合具有不同属性的信息作出决策。正是基于这一思想,在 TCC-EFE^[7]中,作者通过将队列长度、队列长度的瞬时变化量、CWND 字段作为决策信息 3 者结合,构造了一个三维两类的分类器,实现了一个公平性增强的 AQM 机制,取得了极好的公平性。但正如前面所言,TCC-EFE 需要对 TCP 协议进行修改,在数据包 TCP 报头增加 CWND 字段,因此难以在实际网络中部署,缺乏可行性。在本文中,沿用分类器的思想,引入了一个与跳数相关的量 h_i 来提高公平性,并结合平均队列长度一起作为决策信息来判断到达链路的数据包是否被丢弃。通过构造一个二维两类费舍尔线性分类器对 RED 算法进行简单的改进,实现了 FERED 算法。由于 h_i 可通过 IP 数据报头中 TTL 字段信息计算得到,因此 FERED 的实现无需对 TCP/IP 协议进行修改。

首先,以二维两类分类器为例,简单介绍费舍尔线性分类器。如图 1 所示,费舍尔线性分类器的目的是要找出最佳的分类直线,使两类不同的样本点(例如,对于 FERED 来说,样本点的横坐标和纵坐标分别为 h_i 和队列长度)在该直线上的投影最大程度地分离开。此后,对实际中的点,将根据点在分类直线上的投影位置将其归入两类中的一类。从图 1 中可以看出,图 1(a)中的直线分类效果要优于右图。下面简单介绍最佳分类直线的求取(即分类器的构造)步骤,详细的证明过程参见文献[8]。

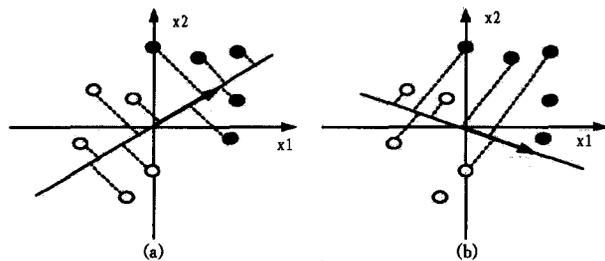


图 1 不同分类直线分类效果示意图

假设两类样本子集 Ω_1, Ω_2 中分别有 n_1, n_2 个样本点,每一个样本点都可以看作是一个二维的列向量 \vec{x} , 该向量以横坐标和纵坐标为两个维度。对于两类样本子集分别定义样本均值向量 \vec{X}_i 和样本离散度矩阵 $\vec{S}_i (i=1,2)$, 如下:

$$\vec{x}_i = \frac{1}{n_i} \sum_{\vec{x} \in \Omega_i} \vec{x}, i=1,2 \quad (1)$$

$$\vec{S}_i = \sum_{\vec{x} \in \Omega_i} (\vec{x} - \vec{X}_i)(\vec{x} - \vec{X}_i)^t, i=1,2 \quad (2)$$

则可得到最佳分类直线方向为:

$$\vec{w} = (\vec{S}_1 + \vec{S}_2)^{-1} (\vec{X}_1 - \vec{X}_2) \quad (3)$$

阈值分界点为

$$w_0 = \frac{\frac{1}{n_1} \sum_{\vec{x} \in \Omega_1} \vec{w}^t \vec{x} + \frac{1}{n_2} \sum_{\vec{x} \in \Omega_2} \vec{w}^t \vec{x}}{2} \quad (4)$$

最终的费舍尔分类器决策规则为:

$$\begin{cases} \vec{w}^t \vec{x} \geq w_0 \rightarrow x \in \Omega_1 \\ \vec{w}^t \vec{x} < w_0 \rightarrow x \in \Omega_2 \end{cases}$$

3 FERED 设计与实现

3.1 利用跳数增强 RED 公平性的动机

在文献[9]中,作者通过大量实际网络中的测量结果得出结论:虽然一个数据流从发送方到达接收方所经历的路由器跳数和其 RTT 之间不存在严格的线性关系,但从总体的趋势上看,数据流所经历的路由器跳数和 RTT 之间大致上还是存在着线性关系的,即数据流经历跳数越多,其 RTT 可能越大。在文献[10]中,作者在实际网络中的测量实验同样得出了近乎相同的结论。据此,我们认为,在路由器的队列管理算法中,适当地保护经过多跳路由器的数据包,在很大程度上意味着保护 RTT 较大的数据流,这将有助于改善由数据流具有不同 RTT 所引发的不公平性问题。

此外,在多拥塞路由器的环境下,数据流的 RTT 中很大一部分将由拥塞路由器处的排队时延决定。经历的拥塞路由器越多,其 RTT 必将相对越大,这将加重数据流具有不同 RTT 所引发的不公平性问题。而且,多拥塞路由器环境下,经历多个拥塞路由器的数据包,其丢弃概率也必将越大,这将使其处于更加不利的位置^[2]。因此,在这种情况下保护经过多跳路由器的数据包,可同时有助于改善由数据流具有不同 RTT 所引发的不公平性问题和由多拥塞路由器所引发的不公平性问题。

正是基于上述的考虑,我们认为可以将数据包经历的路由器跳数作为一个参考量来改善公平性,并据此尝试实现了一个公平性增强的 RED 算法(FERED)。

3.2 二维两类分类器设计

要构造一个二维两类的分类器,首先必须选取所需的状态观测量作为两个维度。队列长度这一状态量作为拥塞观测量,其有效性已经被广泛承认,因此我们也沿用了该状态量作为一个维度。同时,基于 3.1 节的分析,为了增强公平性,我们引入了一个与数据包所经历的路由器跳数相关的状态变量 h_i 作为另一个维度,这一变量可以通过 IP 数据包报头中的 TTL 字段信息计算得到,其计算式是:

$$h_i = v_{hop} / avg_{hop} \quad (5)$$

式中, v_{hop} 为数据包已经经历的路由器跳数,在 NS2 仿真中为 32 与数据包 IP 报头 TTL 字段中的值之差。虽然在实际的网络中,不同操作系统上的发送方所发出的数据包可能具有不同的 TTL 字段初始值,但还是可以区别出来的。因为在常见的操作系统中, TTL 字段初始值一般为 32, 64, 128 和

255,而在实际的网络中极少有数据包经过的路由器数超过 30 跳^[10]。 avg_hop 为到达队列的数据包所经历的平均路由器跳数,按照下式所示的指数加权滑动平均算法(EWMA)计算:

$$avg_hop = avg_hop \times \alpha + v_hop \times (1 - \alpha) \quad (6)$$

式中, $0 < \alpha < 1$, α 为加权因子。这样,当 $h_i < 1$ 时,则认为该数据包经历的路由器跳数相对较少;当 $h_i > 1$ 时,则认为该数据包经历的网关跳数相对较多。因此,当网络处于拥塞状态时,应优先丢弃 h_i 较小的数据包。

基于保护经过多跳路由器的数据包的思想,根据大量试验结果数据,依靠经验选出样本数据,如图 2 所示。

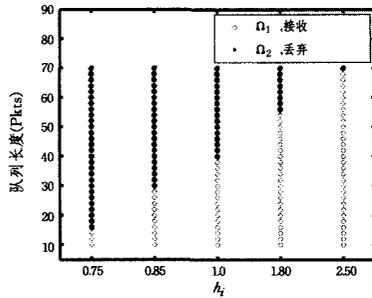


图 2 样本数据

接着再将样本点数据代入式(1)~式(4),即可得到分类器的最佳分类直线方向和阈值分界点分别如下:

$$\vec{w} = (-0.0011 \quad 0.0445)', \vec{w}_0 = 0.0054$$

3.3 FERED 的实现

在 RED 算法中,当每一个数据包到达路由器队列时,该数据包被随机丢弃的概率 p 如下:

$$p = \begin{cases} 0, & Q_e < Q_{\min} \\ \frac{Q_e - Q_{\min}}{Q_{\max} - Q_{\min}} \times p_{\max}, & Q_{\min} < Q_e < Q_{\max} \\ 1, & Q_e > Q_{\max} \end{cases} \quad (7)$$

式中, Q_{\min} 和 Q_{\max} 分别是最小和最大队列长度阈值; Q_e 为当前平均队列长度; p_{\max} 为最大丢弃概率。为了提高 RED 的公平性性能,在第 3.2 节尝试构造了二维两类分类器,并以此对 RED 算法进行简单的修改,修改后随机丢弃概率如下:

$$p = \begin{cases} 0, & Q_e < Q_{\min} \\ 0, & \vec{w}'x \geq \vec{w}_0, \quad Q_{\min} < Q_e < Q_{\max} \\ p_{\max}, & \vec{w}'x < \vec{w}_0 \\ 1, & Q_e > Q_{\max} \end{cases} \quad (8)$$

这样,当 $Q_{\min} < Q_e < Q_{\max}$ 时,用分类器简单取代了原来的算法,使已经经过多跳路由器的数据包得到一定程度的保护。

可以看到,在上述 FERED 算法的实现中,只是用分类器方法对 RED 算法进行了修改,无需对原有的 TCP/IP 协议进行任何修改,因此 FERED 算法实现简单,可行性较好。

4 FERED 性能评价

在 NS2.29^[11] 仿真试验平台上,通过和经典的 RED 算法和 TCC-EFE 机制相比较来评价 FERED 的性能。仿真采用图 3 所示的“停车场”网络拓扑结构,这是多拥塞网关的一种典型配置,其中具有 4 个瓶颈链路。具有不同 RTT 的 TCP 数据流将经历不同数量的拥塞路由器。

为了更真实地模拟实际网络环境,试验中的数据源由 3 类组成:第一类由 N_1 个持久的 FTP 数据流组成;第二类由

N_2 个突发的 HTTP 数据流组成;第三类由 N_3 个非适应性 UDP 数据流组成,其空闲和突发时间周期服从负指数分布,空闲和突发平均时间分别为 10s 和 1s。试验中,让 $N_{11} = N_{21} = N_{31} = N_{41} = n$, $N_{12} = N_{22} = N_{32} = N_{42} = n$, $N_{13} = N_{23} = N_{33} = N_{43} = 2$,通过使 n 的值在 5 到 45 之间变化来模拟链路负载从轻载到重载的变化。为使所有的瓶颈链路具有相同的负载,使 $N_{10} = N_{21} + N_{31} + N_{41}$, $N_{20} = N_{31} + N_{41}$, $N_{30} = N_{41}$ 。模拟中假定所有瓶颈链路的带宽和广播延时分别为 2Mbps 和 10ms,其他链路的带宽和广播延时均为 10Mbps 和 5ms。在 $t=0$ s 时,启动 FTP 和 UDP 数据流,HTTP 数据流随机启动。整个模拟持续 100s。试验中 RED, TCC-EFE 和 FERED 的参数配置如下。

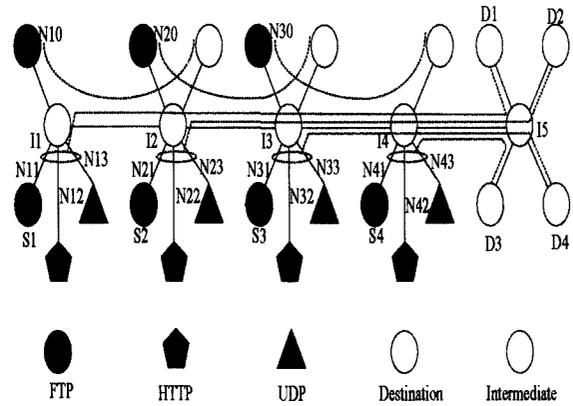


图 3 “停车场”拓扑结构

RED:最大、最小队列长度阈值分别为 75、15 个数据包,其它参数采用默认值。

TCC-EFE:期望队列长度为 20 个数据包,采样间隔为 8ms。最佳分类直线方向和阈值分别为 $\vec{w} = (0.0006 \quad 0.0005 \quad 0.0225)'$, $\vec{w}_0 = 0.0356$, $\alpha = 0.875$ 。

FERED:最佳分类直线方向和阈值分别为 $\vec{w} = (-0.0011 \quad 0.0445)'$, $\vec{w}_0 = 0.0054$, $\alpha = 0.6$,其它参数均采用和 RED 相同的默认值。

首先,在不同的网络负载情况下($n=10, 20, 40$),通过随机跟踪节点 13 上的队列长度来评价 FERED 算法对队列长度的控制能力。其结果如图 4 所示。

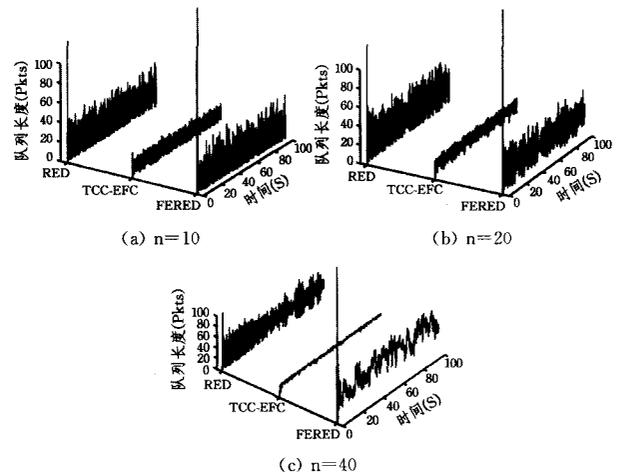


图 4 不同负载下节点 3 上的队列长度

从图 4 可以看到, TCC-EFE 具有最好的队列长度控制能力,这是因为该算法修改了数据包 TCP 报头,增加了 CWND

字段作为显示的通告信息以帮助路由器实现队列管理。同时可以看到,和 RED 算法相比,FERED 保留了 RED 对队列长度控制的优点。当 $n=40$ 时,FERED 甚至可以使队列长度维持在更加稳定的水平。

为了评价 FERED 算法在不同负载情况下的公平性性能,从经历不同数目拥塞链路的 4 组数据源 S1、S2、S3、S4 中分别随机选取一个 FTP 数据流,跟踪它们在不同负载情况下的拥塞窗口的变化。这 4 个 FTP 数据流在 $n=10, 20, 40$ 时的拥塞窗口变化分别如图 5—图 7 所示。

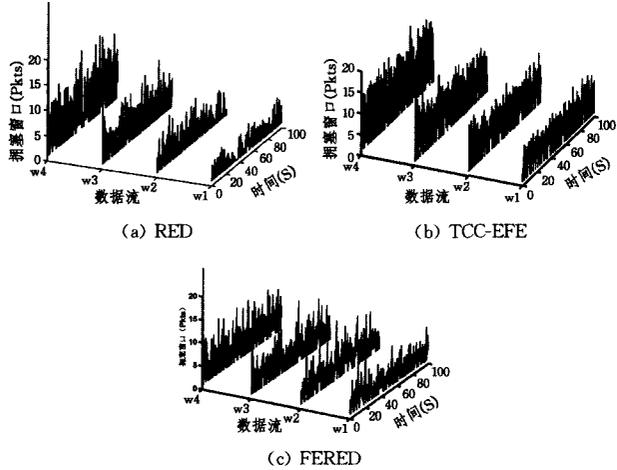


图 5 $n=10$ 时经过不同数量拥塞链路的数据流 CWND

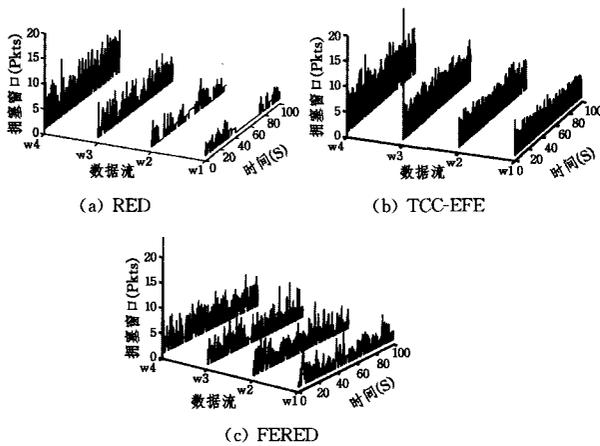


图 6 $n=20$ 时经过不同数量拥塞链路的数据流 CWND

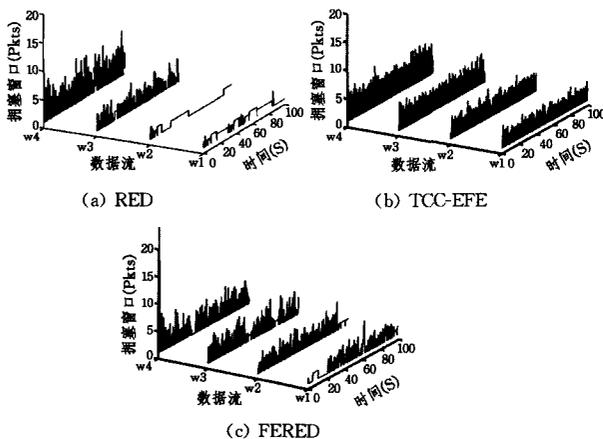


图 7 $n=40$ 时经过不同数量拥塞链路的数据流 CWND

从这些图中可以看到,当 $n=10$ 时,各个链路均处于轻度负载之下,RED 算法也可以取得较好的公平性。但是随着链

路负载增加,当 $n=20$ 时,如图 6 所示,多瓶颈链路环境开始出现,在 RED 算法中经历拥塞链路较多的数据流 w_1 和数据流 w_2 将处于明显的不利位置。随着链路负载继续的增加,当 $n=40$ 时,如图 7 所示,RED 算法中数据流 w_1 和 w_2 几乎出现“饿死”的现象。相比之下,FERED 算法则明显改善了存在于 RED 算法中的不公平性问题,在任何链路负载情况下,都可以取得和 TCC-EFE 相似的公平性,数据流 w_1 和 w_2 始终具有适当大小的 CWND。

在文献[12]中,作者定义了被广泛采用的公平性评价指标

数 FI ,其定义为 $FI = \frac{(\sum_i x_i)^2}{n \sum_i x_i^2}$ 。为了更加清晰、定量地比较 3

种算法的公平性性能,我们借助该公平性指数定义,计算不同负载下各算法的公平性指数,其结果显示在图 8 中。此处 x_i 为 FTP 数据流 w_1, w_2, w_3 和 w_4 成功发送的数据包的数量。

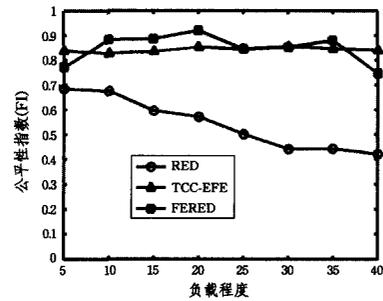


图 8 不同负载下各算法的公平性指数

显然,FERED 算法有效地改善了 RED 算法的公平性性能,取得了和 TCC-EFE 相近的公平性,在链路负载从轻载到重载的变化过程中,FERED 算法的公平性指数大多在 0.8 以上。

结束语 本文基于实际网络中测试结果,提出了利用数据包所经过的路由器跳数来增强 RED 算法公平性的思路,并尝试构造了一个二维两类分类器对 RED 算法进行简单的改进,实现了一个公平性增强的 RED 算法(FERED)。NS2 下的模拟试验表明,FERED 可有效改善公平性,而且实现简单,可行性好。

FERED 算法的缺点是当经过多跳路由器的数据包却具有较小 RTT 时,对这样的数据包的保护将会加重因数据流具有不同 RTT 所引发的不公平性。但是,根据文献[9, 10]在实际网络中的测量实验结果看,这种情况出现的概率是很低的,因此我们认为 FERED 算法还是可行的。

参考文献

- [1] Chiu D, Jain R. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks[J]. Computer Networks and ISDN Systems, 1989, 17(1): 1-14
- [2] Floyd S. Connections with multiple congested routers in packet-switched networks, part 1: one-way traffic[J]. Computer Communication Review, 1991, 21(5): 30-47
- [3] Wei D X, Jin C, Low S H, et al. FAST TCP: Motivation, architecture, algorithms, performance[J]. IEEE/ACM Transactions on Networking, 2006, 14(6): 1246-1259
- [4] Yuan C, Tan L S, Lachlan L H, et al. A generalized FAST TCP-scheme[J]. Computer Communications, 2008, 31(14): 3242-3249

- [5] 王胜灵,侯义斌,黄建辉,等. PFWRR: 能实现比例公平的增强型 WRR[J]. 计算机学报, 2008, 31(5)
- [6] 谢钧,俞璐,金凤林. 基于排队时延和丢包率的拥塞控制[J]. 电子与信息学报, 2010, 32(9)
- [7] Ren F Y, Huang X M, Lin C. Design of a Two-category Classifier with Multi-dimension for Active Queue Management[C]// First ACM SIGCOMM Asia Workshop. Beijing, China, April 2005: 150-157
- [8] 边肇祺,张学工,等. 模式识别(第二版)[M]. 北京:清华大学出版社, 2000: 87-90
- [9] Fei A, Pei G, Liu R, et al. Measurements on delay and hop-count of the internet[C]// IEEE GLOBECOM'98-Internet Mini-conference. Sydney. Australia, November 1998
- [10] Fujii K, Goto S. Correlation between hop count and packet transfer time[C]// APAN/IWS2000. February 2000: 15-17
- [11] UCN/LBL/VINT. Network Simulator-NS2 [EB/OL]. <http://www.mash.cs.berkeley.edu/ns>
- [12] Jain R, Chiu D M, Hawe W. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems [R]. DEC-TR-301. September 1984

(上接第 61 页)

度较大的网络中接近 Combo。参数 $K=10$ 时, 链路信道数目大于 $K=6$ 时的链路信道数目, 这是因为经过更多的循环, 簇结构更为优化。但前者的时间开销是后者的 1.57 倍。

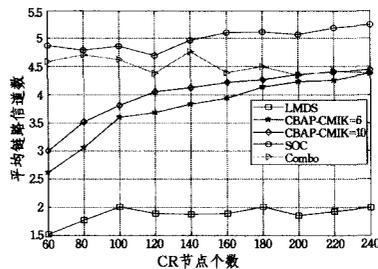


图3 平均链路信道数

图4给出了网络中平均簇内公共信道数目的比较。由此可见, CBAP-CMI算法获得的平均簇内公共信道数目大于3, 接近以优化簇内公共信道数目为目标的SOC算法和Combo算法; LMDS算法不能保证簇内有公共信道。此外, 增加消息交互次数可以缓慢增加簇内公共信道数。

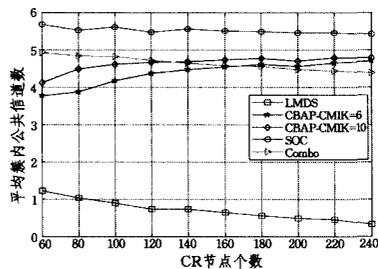


图4 平均簇内公共信道数

综合上述仿真结果可知, CBAP-CMI算法产生的簇数目更少, 获得的平均链路可用信道数目和簇内公共信道数目接近Combo等优化簇内公共信道的分簇算法。增加CBAP-CMI消息交互次数可以以较大的时间开销缓慢增加簇内公共信道数目和链路信道数目。

结束语 提出了一种基于AP算法模型的认知Ad-hoc网络分簇算法, 通过认知用户的分布式消息交互, 建立起相对稳定、高效的簇结构。仿真结果显示, 该算法减小了网络中的簇数目, 提高了簇内通信平均可用链路数目, 保证了簇内节点至少有一个公共信道, 减小了控制开销, 也提高了簇结构的稳健性, 为认知Ad-hoc网络中分布式频谱协作提供了高效的网络拓扑结构。

参考文献

- [1] Spectrum policy task force report [R]. 022135. Federal Communications Commission, Nov. 2002

- [2] Akyildiz I F, Lee W-Y. NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey[J]. Computer Networks, 2006, 50: 2127-2159
- [3] XG Working Group. The XG Vision RFCs [SB/OL]. <http://www.ir.bbn.com/projects/xmac/rfc/rfc-vision.pdf>
- [4] End-to-End Reconfigurability. E2R_WP5_D5_3_050727 [RB/OL]. http://e2r.motlabs.com/Deliverable/E2R_WP5_D5_3_050727.pdf
- [5] Zhao Jun, Zheng Hai-tao, Yang Guang-hua. Distributed Coordination in Dynamic Spectrum Allocation Networks[C]// IEEE Dyspan05. 2005: 259-268
- [6] Akyildiz I F, Lee W-Y, Chowdhury K R. CRAHNS: Cognitive radio ad hoc networks[J]. Ad-hoc Networks, 2009, 7(5): 810-836
- [7] Komali R S, Thomas R W, Dasilva L A, et al. The Price of Ignorance-distributed Topology Control in Cognitive Networks[J]. IEEE Transactions on Wireless Communications, 2010, 9(4)
- [8] Ren W, Zhao Q, Swami A. On the Connectivity and Multihop Delay of Ad-hoc Cognitive Radio Networks[Z]. arXiv: 0912.4087v1
- [9] Yu J Y, Chong P H J. A survey of clustering scheme for mobile Ad-hoc networks[C]// IEEE Communications Surveys & Tutorials. First Quarter 2005
- [10] Zhang C T, Maggio H, Chlamtac G M, et al. Topology Management in CogMesh: A Cluster-based Cognitive Radio Mesh Network[C]// IEEE ICC2007 Proceedings. June 2007: 6516-6521
- [11] Lazos L, Liu Si-si, Krunz M. Spectrum opportunity-based control channel assignment in cognitive radio networks[C]// IEEE Secon2009 Proceedings. 2009
- [12] Asterjadhi A, Baldo N, Zorzi M. A cluster formation protocol for cognitive radio ad hoc networks[C]// 2010 European, Wireless Conference(EW). 2010: 955-961
- [13] Baddour K E, Ureten O, Willink T J. Efficient clustering of cognitive radio networks using affinity propagation [C]// IEEE Crowncom2009 Proceedings. 2009
- [14] Frey B, Dueck D. Clustering by passing messages between data points[J]. Science, 2007, 315: 972-976
- [15] 许文竹, 徐立鸿. 基于仿射传播聚类的自适应关键帧提取[J]. 计算机科学, 2010, 37(1): 268-270
- [16] Sumedha M L, Weigt M. Clustering by soft-constraint affinity propagation: Applications to gene-expression data[J]. Bioinformatics, 2007, 23(20): 2708-2715
- [17] Krishnamurthy S, Thoppian M, Kuppa S, et al. Time-efficient distributed layer-2 auto-configuration for cognitive radio networks[J]. Computer Networks, 2008, 52(4): 831-849
- [18] Arachchige C L, Venkatesan S, Mittal N. Asynchronous Neighbor Discovery Algorithm for Cognitive Radio Networks[R]. UTDCS-24-08. University of Texas at Dallas, August 2008