

Research on the Fair Assignment of Router's Bandwidth

Zifen Yang

School of Information, Linyi normal University

Shuang ling Street, Linyi 276000, China

E-mail: yzf_smile@163.com

Abstract

The problem of providing a fair allocation between responsible flows (such as TCP) and non-responsible (such as UDP) is investigated. A core stateless fair queuing algorithm with high throughput is presented. In the algorithm, the same packet loss strategies that are based on the calculated probability p are used to all of the flows without distinction. And it does not make into account the sensitivity of TCP. So this paper proposes a new algorithm VCSFQ do the different packet loss strategy. The algorithm can protect TCP flows against UDP flows and improves the fairness of bandwidth allocation. The algorithm performs better than CSFQ for getting high throughput. Simulation shows that it is effective and efficient.

Keywords: TCP flows, UDP flows, CSFQ, Fair assignment of bandwidth

1. Introduction

The flow of Internet is a mixture of different kind of data flows, which include TCP flows, TCP kind of flows, non-TCP friendly flow, and non-responsible flows and so on. Because the conquer of non-responsible flows and non-TCP friendly flow to shared resource, hindering the fair assignment of bandwidth between normal TCP flow and kindly TCP flow, and has a bad influence on the Internet. Therefore, it will have a magnificent influence on the service quality of Internet, if we arrange different kind of flow separately and realize the fair share of bandwidth.

2. Related work

It is indicated by most research that there are still exist three kind of problems hindering promote of service quality .The first problem is the fair assignment between TCP flow and UDP flow. The second is the fair assignment of bandwidth between the separate flows among the aggregate flow; the third problem is the fair assignment of the left bandwidth between aggregate flows. Since the jam of the TCP flow and the slow mobility, TCP flow is extreme sensitive to the jam and data dropping. However, UDP flow is not influenced by that, so when the Internet is jammed, the delivery rate of TCP data will decline extremely. At the same time, UDP flow will take place a lot of Internet resource, which prevent TCP flow from getting fair assignment of Internet resource.

3. CSFQ Algorithm

3.1 Overview

The good of CSFQ algorithm is to arrange the bandwidth in shared Internet flow fairly and support endowing a weight to each flow, the distribute bandwidth base on weight. When package p arrives, edge rather classify it, then decide which $flow_i$ each of them belonging to, then estimate their arrival rate AR_i . The estimation of $flow_i$ arrive rate is as follows:

$$AR_{i,new} = (1 - e^{-\frac{T}{K}}) \frac{L}{T} + \frac{T}{K} AR_{i,old} \quad (1)$$

In this formula, $AR_{i,old}$ is a roughly character of the former arrival rate of flow; K is a constant; L is the length of the bag P ; T is the interval time between the former bag and current bag, $AR_{i,old}$ will be the label of bag P .

When bag P arrives, edge and centre rather will use formula (2) to calculate the rate of dropped bags, then transmit or desert data bags.

$$P_{drop} = \max(0, 1 - \frac{FS}{P.label}) \quad (2)$$

In this formula, FS is a fair shared rate calculated when the former bag arrived, $P.label$ is the label of current bag. Obviously, when $P.label > FS$, $P > 0$; when $P.label \leq FS$, $P = 0$.

So, when $P.label > FS$, package P will be dropped at the rate of P . When there is another bag come in, a re-calculation of the fair shared rate will be needed. Firstly, we will use formula (1), by even the index, estimate the arrival rate AR of aggregate flow in rather and the rate FR of transmitted, then reserve buffer. The management of buffer is by means of "dropping bottom", if buffer is full, then the arriving data bags will be dropped.

3.2 A new improvement of CSFQ

Based on the research of CSFQ, we find that the dropping tactics of CSFQ is based on non responsible flow, the "dropping bag" tactics is applied into all flow at the rate of P_{drop} , without identify non responsible flow, neither consider about the sensitive and reactive of TCP. For TCP flow, dropping one group means slow down half of the transmitting rate. Besides, since the characteristic of data flow, such as UDP flow will get extra bandwidth that TCP flow has lost. Then the fair assignment of bandwidth will be damaged.

Because the reason we have mentioned above, we will change the CSFQ algorithm. Here we introduce a new parameter V , the variable V is the ideal character of the average rate of TCP flow. If we want the average rate of TCP flow reach the ideal character V , we should put off the time of TCP flow dropping bags. That is to say, when the Internet is jammed, the system don't drop TCP flow immediately, instead dropping it when its rate reach $4/3 V$, when its rate wave between $2/3 V$ and $4/3 V$, so we get the ideal average rate. Thus, rather should estimate the ideal average rate V accurately. Based on the principle of dropping bags, the rate that UDP flow got will almost equal to a . If we can estimate the average rate of an accurately, then we can get the average rate of UDP flow. With this rate to require TCP flow, then we can balance TCP flow and UDP flow.

By the logic we have mentioned above, we will improve it like this, carry out different tactics according to TCP flow and UDP flow. We have improved formula (2) and have designed the means of dropping bags as follows. For UDP flow, the tactics is:

$$P_{UDP} = \max[0, 1 - \frac{\alpha(t)}{r_i(t)}] \quad (3)$$

For TCP flow, the tactics is:

$$P_{TCP} = \begin{cases} 0, & r_i(t) < \frac{4}{3}V \\ \max[0, 1 - \frac{a(t)}{r_i(t)}], & r_i(t) \geq \frac{4}{3}V \end{cases} \quad (4)$$

As have mentioned above, the arrival rate of each flow is $r_i(t)$, the fair shared rate is $a(t)$ at t time.

3.3 The Analysis and experiment on Improved Algorithm

3.3.1 Analysis

In the primitive CSFQ, when data bag P arrive, the edge rather will classify it, decide it belong to which flow, then suppose it is flow, after that, estimate the arrival rate of this flow. Nevertheless, when bag P arrive, the edge and centre rather will transmit data bags according to the dropping bags rate calculated by the formula we have mentioned above.

In V-CSFQ, the calculated of arrival rate and average rate of data bags is as same as CSFQ; follow the formula (1) and (3). When arrival rate and average rate, that is to say when jam happened, for different data flow. TCP flow and UDP flow should resort different algorithm. TCP flow should follow formula (5) and UDP flow should follow formula (4), so that will be fair to both TCP flow and UDP flow.

Compared with CSFQ, V -CSFQ is as complicated as it; both the calculation of arrival rate and average shared rate is as same as the former. What need to do is to carry out different drop algorithm according to different flow. In drop algorithm, the drop time was postponed and the transmutation rate was promoted by linear.

3.3.2 Experiment

Now, let's imitate and experiment the function of the V -CSFQ in NS. Algorithm in the situation of TCP flow and UDP flow share the jammed link.

In the experiment, TCP flow and UDP flow share one jammed link, the import rate of link is 10Mb/s, the cache size is 128Kbyte, UDP is transmitted at the rate of 10Mb/s from start to bottom, TCP flow's RTT is 100ms. It takes fast retransmit and recorder as tactics to cope with the amount of dropped bags which according to the same agreement and transmission principle as the web transmission system in reality. Tragedy K (the interval time used for estimating the arrival rate of each flow), K_a (the interval time used for estimating the fair shared rate) and K_c (the interval time used for judging whether the link is jammed or not) are 200ms, the text time is 30ms. We use the topology graph show as Figure 1.

Insert Figure 1 Here

Figure 1 is the ratio that the throughput in reading to ideal throughput. The algorithm of ideal throughput can get through data link rate multiply text time then divide the amount of flow.

We consider about three kinds of situations. First the data which will be sent are a TCP flow and seven UDP flows. Second, the data which will be sent are four TCP flows and four UDP flows. Third, the sent data are seven TCP flows and one UDP flow. Then we can get table I by the simulate result.

Insert Table 1 Here

The most web transmission in reality belongs to the third situation. In other words, TCP flow is the majority in most situations and a few UDP flow appear occasionally. It is obvious in the text result that the TCP flow gets less bandwidth than UDP flow in the CSFQ algorithm. However, according to the figures in the chart, TCP flows get more bandwidth than UDP flow in the improved algorithm. Through TCP flow get less in chart three, get they are almost same. Then we can see clearly now that in all kind of situations, the TCP flow and UDP flow get more fair assignment of bandwidth than the primitive algorithm.

4. Conclusion

V-CSFQ carries out different kind of drop tactics, and makes sure the fair assignment between UDP flow and TCP flow. The simulate result have proved that V-CSFQ has an advantage in bandwidth assignment than CSFQ. Besides, drop groups, combine and scheduling algorithm is not only another access to improve the web fair, but also a research direction to improve QOS.

References

- [Online] Available: <http://www.w3.org/TR/2002/CR-Xmlesc-corene-20020304/>,2003-07.
- Li, Shenglei, Zhang, Dejun & Liu, Gang. (2006). A New Algorithm Of Bandwidth Assignment. *Newspaper of Xi An Transportation University*, 85, 123-126.
- Wu, Jianpiang & Xu, Xiaomei. (2008). *Advanced Computer web* (1st ed.). Machinery Industry Press. Beijing (Chapter 4).
- Xie, Xiren. (2005). *Computer Network* (3rd ed.) Electronic Industry Press. Beijing (Chapter 3).
- Zhang, Mingxin & Chen, Guohang. (2007). The Analysis And Improvement of CSFQ Algorithm *Micro-computer System*, 186, 28-31.

Table 1. The result of the experiments

	1	2	3	4	5	6	7	8
CSFQ	0.55	0.92	0.95	0.90	0.88	0.92	0.90	0.95
V-CSFQ	0.95	0.87	0.85	0.86	0.88	0.85	0.86	0.87
V-CSFQ	0.95	1.15	0.95	0.90	0.93	0.90	0.90	0.92

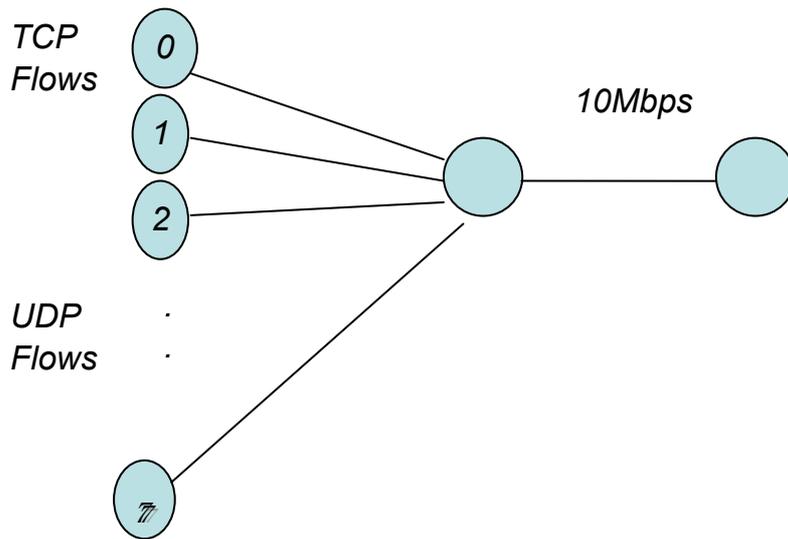


Figure 1. Topology graph of this experiment