IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

Congestion Control for Peer to Peer Application using Random Early Detection Algorithm

Sonam

PG Student, Department of Computer Science, United College of Engineering & Research Allahabad, India

Abstract: This paper presents an extension of Random Early Detection Algorithm to avoid congestion in the peer to peer connection. Avoid congestion either by dropping the packet or arriving at the gateway. In this algorithm we use multiple queues for buffering and storing the packets one is the main queue for calculating the average queue length and use multiple threshold value for each queue. RED gateways keep the average queue size low while allowing occasional bursts of packet in the queue during congestion. In this way we can avoid the congestion.

Keywords: Peer to Peer network, RED Algorithm, congestion, multiple queues.

1. INTRODUCTION

An easy way to Peer-to-peer (P2P) is network model to that provided by traditional client-server architecture. P2P networks use a model in which each machine, referred to as a node, basically node means peer and it works as a client with its own layer of server functionality. A peer acts the role of a client and a server concurrently. That is, the peer can initiate requests to other peers, and at the same time respond to incoming requests from other peers on the network. It differs from the traditional client-server model where a node can only send requests to the other node and then wait for the server's response. With a client-server approach, the performance of the server will become worse as the number of clients requesting services from the server increase.

However, in P2P networks overall network bandwidth actually improves as add more number of nodes are added to the network. These nodes can organize themselves into ad-hoc groups as they communicate; work together and share bandwidth with each other to complete the tasks at hand (e.g. file sharing). Each peer can upload and download concurrently, and in a process like this, new peers can join the group while earliest peers leave at any time. This active re-organization of group peer members is transparent to end-users. In the peer to peer network at a time many users can send and receive information concurrently there is no router required for this type of connection, for enhancing the bandwidth of the peer to peer connection we will use RED algorithm so that there is minimum loss of packet occur and we get maximum utilization of the network . For avoiding the congestion in peer to peer connection instead of using one queue we will use multiple queues for buffering and storing the data. Random Early Detection Gateways for congestion avoidance in packet-switched networks. The gateway detects in early stage of congestion by calculating the average queue size. The gateway could inform connections of congestion either by dropping packets arriving at the gateway or by setting a bit in packet headers. When the average queue Size beyond a Preset threshold, the gateway drops or marks each arriving packet with a certain being probable, where the exact Probability is a function of the average queue size. RED gateways remain the average queue size low while allowing infrequently bursts of packets in the queue. Throughout congestion, the probability that the gateway inform a particular connection to reduce its window is roughly proportional to that connection's share of the bandwidth from beginning to end the gateway.

RED gateways can be efficient in gateways with a range of packet-scheduling and packet-dropping algorithms. For Example, RED congestion control structure could be Implemented in gateways with drop being preferred, where packets are marked as either "indispensable" or "optional", and "optional" Packets are dropped first when the queue more than a certain size.

2. PREVIOUS WORK

In the previous work several researchers have studied Early Random Drop gateways as a method for providing congestion avoidance at the gateway.1 Hashem [11] gave some of the deficiency of Random Drop2 and Drop Tail gateways, and briefly examine Early Random Drop gateways. In the implementation of Early Random Drop gateways in [11], if the Queue length more than a certain drop level, and then the gateway drops each packet arriving at the gateway with a fixed drop probability. This is detailed as a rough initial implementation.

Hashem [11] stresses that in future implementations the drop level and the drop probability should be adjusted dynamically, depending on network traffic. Hashem [11] points out that with Drop Tail gateways each congestion period introduces global synchronization in the network. When the queue flows over, packets are often dropped from several connections, and these connections decrease their windows concurrently. This results in a loss of

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

throughput at the gateway. The paper shows that Early Random Drop gateways have a broader view of traffic distribution than do Drop Tail or Random Drop gateways and become smaller global synchronization. The paper evoke that because of this broader view of traffic distribution, Early Random Drop gateways have a better chance than Drop Tail gateways of targeting aggressive users. The conclusions in [11] are that Early Random Drop gateways deserve further inquires. For the version of Early Random Drop gateways used in the simulations in [36], if the queue is more than half full then the gateway drops each arriving packet with probability 0.02. Zhang [36] shows that this version of Early Random Drop gateways was not having success in controlling Behave badly users. In these simulations, with both Random Drop and Early Random Drop gateways, the misbehaving users received roughly 75% higher throughput than the users implementing standard 4.3 BSD TCP. The Gateway Congestion Control Survey [21] considers the versions of Early Random Drop described above. The survey cites the results in which the Early Random Drop gateway is unsuccessful in controlling misbehaving users [36]. As mentioned in [32], Early Random Drop gateways are not expected to solve all of the problems of unequal throughput given connections with different roundtrip times and multiple congested gateways. In [21],the goals of Early Random Drop gateways for congestion avoidance reappearances as "uniform, active treatment of users (streams/flows), of low overhead, and of good scaling characteristics in large and loaded networks". It is left as an open question whether or not these goals can be achieved. One more approach in the previous paper which avoids congestion in some other way early descriptions of IP Source Quench messages suggest that gateways could send Source Quench messages to source hosts before the buffer space at the gateway Reaches capacity [26], and before packets have to be dropped at the gateway. One proposal [27] suggests that the gateway send Source Quench messages when the queue size exceeds a certain threshold, and outlines a possible method for flow control at the source hosts in response to these messages. The proposal also evoke that when the gateway queue size nearer the maximum level the gateway could discard arriving packets other than ICMP packets. The DEC bit congestion avoidance scheme, a binary response scheme for congestion avoidance, is described in [29]. In the DEC bit scheme the gateway uses a congestion indication bit in packet headers to provide response about congestion in the network. When a packet arrives at the gateway, the gateway calculates the average queue length for the last (busy + idle) period plus the current busy period. (The gateway is busy when it is transmitting packets, and idle otherwise.) When the average queue length more than exceeds, then the gateway sets the congestion-suggestive bit in the packet header of arriving packets. The source uses window flow control, and brings up to date its window once every two roundtrip times. If at least half of the packets in the last window had the congestion indication bit set, then the window is decreased exponentially otherwise, the window is increased linearly. There are several significant differences between DEC bit gateways and the RED gateways described in this paper.

The first difference concerns the method of calculating the Average queue size. Because the DEC bit scheme choose the last (busy + idle) cycle plus the current busy period for averaging the queue size, the queue size can sometimes be averaged over a rather short period of time. In high-speed networks with restively large size buffers at the gateway, it would be desirable to stated control the time constant for the computed average queue size; this is done in RED gateways using time-based exponential decline. In [29] the authors report that they rejected the idea of a weighted exponential running average of the queue length because when the time interval was far from the roundtrip time, there was bias in the network. This problem of bias does not arise with RED gateways because RED gateways use a randomized algorithm for marking packets, and assume that the sources use a separate algorithm for act to marked packets. In a DEC bit network, the source looks at the fraction of packets that have been marked in the last roundtrip time. For a network with RED gateways, the source should reduce its window even if there is only one marked packet.

3. PROPOSED WORK

The objective is to provide the congestion control avoidance algorithm which avoid the congestion as early as possible either by dropping the packet or forwarding the packet in this way we can minimize the congestion, before explaining the proposed work first I will give a small introduction about RED algorithm, In the RED algorithm on the basis of the average queue length we will forward or drop the packet. In my proposed I have taken multiple queues for storing and buffering the data, pass the packet to the main queue and set the multiple threshold value for each queue now on the basis of this threshold value we further forward or drop the packet.

4. PROPOSED ALGORITHM

Input's// n number of queues
$Th_{1,max}, Th_{2,max}, \dots, Th_{n,max}$ //Maximum threshold for each queue
$Th_{1,min}, Th_{2,min}, \dots, Th_{n,min}$ //Minimum threshold for each queue

Event: A new packet arrives at some queues (say k out of n)
Compute the average queue lengths $\bar{q}_i \ \forall i = 1,2,\cdots,n$.
For queue 1 to k **do**
{
If ($\bar{q}_i \leq Th_{i,min}$)
{
Forward packet;
}
Else if ($Th_{i,min} \leq \bar{q}_i \leq Th_{i,max}$)

DOI 10.17148/IJARCCE.2016.5807

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

{
Forward the packet with probability p;
}
Else ($\bar{q}_i > Th_{i,max}$)
{
Drop the packet;
}

Compute average size of main queue $\bar{q} = \frac{1}{n}\sum_{i=1}^{n}\bar{q}_i$
Set thresholds for main queue Th_{max} and Th_{min}
If ($\bar{q} \leq Th_{min}$)
{
Forward packet;
}
Else if ($Th_{min} \leq \bar{q} \leq Th_{max}$)
{
Forward the packet with probability p;
}
Else ($\bar{q} > Th_{max}$)
{
Drop the packet;
}
Remain idle until $\bar{q} \leq Th_{min}$

5. DESIGN GUIDLINES

This section summarizes some of the design goals and guidelines for RED gateways. The main goal is to provide congestion avoidance by controlling the average queue size. Additional goals include the avoidance of global synchronization and of a bias against busty traffic and the ability to maintain an upper bound on the average queue size even in the absence of cooperation from transport layer protocols. The first job of a congestion avoidance mechanism at the gateway is to detect incipient congestion. As defined in [18], a congestion avoidance scheme maintains the network in a region of low delay and high throughput. The average queue size should be kept low, while network with connections with a range of roundtrip times, throughput requirements, and delay sensitivities, the gateway is the most suitable agent to determine the size and duration of fluctuations in the actual queue size should be allowed to accommodate bursty traffic and transient congestion. Because the gateway can monitor the size of the queue over time, the gateway is the suitable agent to detect early stage congestion.

Because the gateway has a unified view of the various sources contributing to this congestion, the gateway is also the appropriate agent to decide which sources to notify of this congestion. In short-lived bursts in queue size to be provided by the gateway.The gateway can do this by controlling the time constants used by the low-pass filter for calculating the average queue size. The goal of the gateway is to detect beginning congestion that has continued for a “long time” (several roundtrip times). The second job of a congestion avoidance entry is to decide which connections to inform of congestion at the gateway. If congestion is detected before the gateway buffer is full, it is not essential for the gateway to drop packets to inform sources of congestion. In this paper, we say that the gateway marks a packet, and notifies the source to become smaller the window for that connection. This marking and notification can composed of dropping a packet, setting a bit in a packet header, or some other procedure understood by the transport protocol. The current response mechanism in TCP/IP networks is for the gateway to drop packets,

6. COMPARISION WITH PERIVIOUS

In the previous work there is only single queue which stores data and also calculate the average queue length , only two threshold value thmax and thmin were use on the basis of this value we forward and drop the packet , but in my proposed work instead of using single we have used more than one queue , for these queue we have taken multiple threshold value now these queue forward the packets to the main queue basis on the some threshold value now main queue calculate the average queue length and basis on the these threshold value we will forward and drop the packet.

- Complexity of Estimation and Iterative Planning
- Complexity in Tool Selection
- Congestion Occurrence.
- There is difficulty for storing and buffering the data
- It leads to retransmit synchronization
- Possibility of losing important information because of congestion.

7. CONCLUSION

In today scenario in the field of networking bandwidth utilization is most concern issue ,Many research have done in the field of networking for utilization of the bandwidth and many research is running .There are so many algorithms and ideas have given by researchers in this field for utilizing the bandwidth now the main concept behind this is that how to increase the network performance so the simple answer is that , avoid to drop the packets ,that means to avoid the congestion is the most appropriate, way is to get the best network performance , as soon as the congestion is minimize there is no delay of packets in the network . now come to the point peer to peer congestion control , in the peer to peer network one node can play the role of client and server at the same time and can transfer and receive the data at the same time ,and there is no central hub require for that process, now when congestion occurs here so from the previous paper I get the idea how to minimize it that is by using RED Random Early Detection algorithm, to calculate the average queue length and set the threshold value for queue on the basis of these values we will forward or drop the packets. Now in my proposed work I have taken multiple queue instead of single queue , the role of these multiple queue is to store and buffer data and set multiple threshold for these queue , now these multiple queue forward the data to the main queue on the basis of these threshold value now main queue calculate the average queue length and compare it with threshold values and on the basis of these values it can forward or drop the packets in this way we can minimize the congestion as early as possible . We can get best network utilization, so there is no loss of packets during data transfer in the network automatically bandwidth will increase. So in this way we can achieve our target for utilizing the bandwidth and there is no load on network, there is no delay of packets as well.

REFERENCES

[1] Bacon, D,. Dupuy, A., Schwartz, J., and Yemimi, Y., "Nest: a Network Simulation and Prototyping Tool", Proceedings of winter 1988 USENIX Conference, 1988, pp. 17-78.
[2] Bala, K., Cidon, I., and Sohraby, K., "Congestion Control for High Speed Packet Switched Networks", INFOCOM '90, pp. 520-526, 1990.
[3] Carta, D., "Two Fast Implementations of the 'Minimal Standard' Random Number Generator", Communications of the ACM, V.33 N.1, January 1990, pp. 87-88.
[4] Clark, D.D., Shenker, S., and Zhang, L., "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", SIGCOMM '92, August 1992, p. 14-26.
[5] Floyd, S., Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1:One-way Traffic, Computer Communication Review, V.21 N.5, October 1991, pp. 30-47.
[6] Floyd, S., "Issues in Flexible Resource Management for Datagram Networks", Proceedings of the 3rd Workshop on Very High Speed Networks, March, 1992.
[7] Floyd, S., and Jacobson, V., On Traffic Phase Effects in Packet-Switched Gateways, Internetworking: Research and Experience, V.3 N.3, September 1992, p.115-156
[8] Floyd, S., and Jacobson, V., The Synchronization of Periodic Routing Messages, to appear in SIGCOMM 93.
[9] Hansen, A Table of Series and Products, Prentice Hall, Englewood Cliffs, NJ, 1975.
[10] Harvey, A., Forecasting, structural time series models and the Kalman filter, Cambridge University Press, 1989.
[11] Hashem, E., "Analysis of random drop for gateway congestion control", Report LCS TR-465, Laboratory for Computer Science, MIT, Cambridge, MA,1989, p.103.
[12] Hoeffding, W., Probability Inequalities for Sums of Bounded Random Variables, American Statistical Association Journal, Vol. 58, March 1963, p. 13-30.
[13] Hofri, M., Probabilistic Analysis of Algorithms, Springer-Verlag, 1987