Volume : 2 | Issue : 4 | April 2013 • ISSN No 2277 - 8160

Research Paper | Computer Science

A study on 'Network Congestion Control Algorithms'

Vidyasagar S.D. IV Sem MCA, JNN Engineering College, Shimoga-577201, Karnataka .

Dr. VidyaShankar M.H. Asst. Professor of Computer Science, Sahyadri Science College, Kuvempu University, Shimoga, Karnataka.

ABSTRACT

The present day networks are meant for transmission of data, voice, multimedia applications, using packet switching and frame relays. With explosion in traffic, the earlier problem of proper routing, are proving to be lacking in a providing acceptable bonds of delay and errors, as network getting jammed due to traffic The current circuit switched and packet switched networks are suffering from the problem of extreme congestion and network crashing frequently. The demand for fast transfer of large volumes of data,. TCP, does not meet this demand because it favors reliability over Timeliness and fails to fully utilize the network capacity due to limitations of its conservative congestion control algorithm. The slow response of TCP in fast long distance networks leaves sizeable unused bandwidth in such networks. A large variety of TCP variants have been proposed to improve the connection's throughput by adopting more aggressive congestion control algorithms. This article deals with different types of congestions algorithms and their roles in network.

KEYWORDS: Hybla,Vegas,Tahoe

Introduction:

When too many packets are present in the subnet, performance degrades and it's called as congestion. Some of the factors that Causes Congestion they are , Even slow processors can also cause for congestion ,Packet arrival rate exceeds the outgoing link capacity,Insufficient memory to store arriving packetsand Bursty traffic. Congestion is like largebandwidth-delay product, Speed mismatch, Topology, Increased Usage, Misbehavior of sources. When the number of packets dumped into the subnet by the hosts is within its carrying capacity, they are all delivered (except for a few that are afflicted with transmission errors) and the number delivered is proportional to the number sent. However, as traffic increases too far, the routers are no longer able to cope and they begin losing packets. This tends to make matters worse.

Congestion control algorithm is the set of instructions to control the congestion. Congestion algorithm operates by sending a message back to the various sources telling them to slow down when the network gets into trouble. Thus a host can get a "slow down" message either because the receiver cannot handle the load, or because the network cannot handle it.In the late 80s, the Internet suffered a congestion collapse. The innovative techniques of TCP Vegas have been the subject of much debate in recent years. Several studies have reported that TCP Vegas provides better performance than TCP Reno.

Traffic management in computer networks:

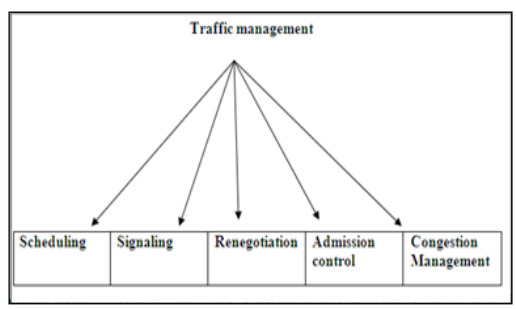
According to Keshav, "Traffic management is a set of policies and mechanisms that, allow a network to efficiently satisfy a diverse range of service request. The fundamental aspect of traffic management is diversity in user requirements and efficiency in satisfying them."

Network Congestion Monitoring system comprises of TCP Vegas, which is a new design for TCP that includes a modified retransmission strategy (compared to TCP Reno) that is based on

fine-grained measurements of the round-trip time (RTT) as well as new mechanisms for congestion detection during slow-start and congestion avoidance. This paper takes a fresh look at the design of TCP Vegas and attempts to shed light on the advantages (and disadvantages) of the innovations introduced by TCP Vegas. In the following paragraph, a solution is explained to handle congestion.

Random Early Detection:

It is well known that dealing with congestion after it is first detected is more effective than letting it gum up the works and then trying to deal with it. This observation leads to the idea of discarding packets before all the buffer space is really exhausted. A popular algorithm for doing this is called RED (Random Early Detection) (Floyd and Jacobson, 1993). The following network represents various modes of traffic management.



Scheduling: Given a set of requests in the service queue, a server uses a scheduling discipline to decide which request to serve next. Scheduling disciplines are important because they are key to fairly sharing network resources and to providing performance critical applications

Signaling: Signaling is the process by which an end point requests the network to setup, tear down, or renegotiate a call.

Renegotiation: The guaranteed-service connection must specify its traffic descriptor at the time of connection establishment.

Admission control: The signaling network carries signaling messages and makes these reservations.

Congestion Management. A network can be said to control congestion if it provides each user with mechanisms to specify and obtain their desired quality of service from the network

Failures of packet conservation:

There are only three ways for packet conservation to fail:

1. The connection doesn't get to equilibrium, or
2 A sender injects a new packet before an old packet has exited, or
3. The equilibrium can't be reached because of resource limits along the path.

Basic Algorithms for congestion control

1. The Leaky Bucket Algorithm: The host is allowed to put one packet per clock tick onto the network.The Leaky Bucket

Algorithm used to control rate in a network. It is implemented as a single-server queue with constant service time. If the bucket (buffer) overflows then packets are discarded.The leaky bucket enforces a constant output rate (average rate) regardless of the burstiness of the input. Does nothing when input is idle.
The host injects one packet per clock tick onto the network. This results in a uniform flow of packets, smoothing out bursts and reducing congestion.When packets are the same size (as in ATM cells), the one packet per tick is okay. For variable length packets though, it is better to allow a fixed number of bytes per tick. E.g. 1024 bytes per tick will allow one 1024-byte packet or two 512-byte packets or four 256-byte packets on 1 tick.

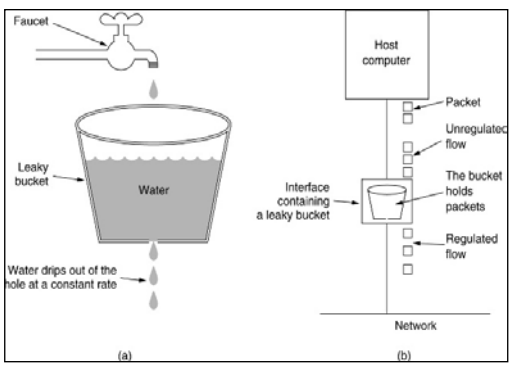


Figure: (a) A leaky bucket with water.
(b) A leaky bucket with packets.

2. The Token Bucket Algorithm: In contrast to the leaky bucket, the Token Bucket Algorithm (TB), allows the output rate to vary, depending on the size of the burst. In the TB algorithm, the bucket holds tokens. To transmit a packet, the host must capture and destroy one token.Tokens are generated by a clock at the rate of one token every Δt sec.Idle hosts can capture and save up tokens (up to the max. size of the bucket) in order to send larger bursts later.

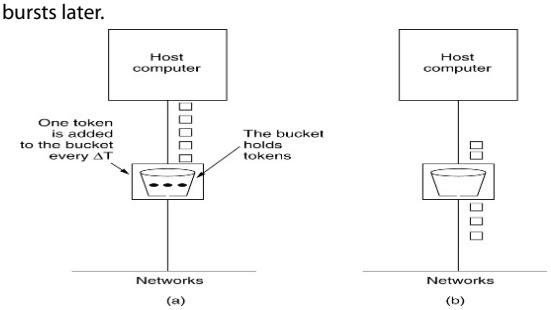


Fig:The Token Bucket Algorithm(a) Before. (b) After

Leaky Bucket vs Token Bucket

a. Leaky Bucket discards packets where are Token Bucket does not.
b. Token Bucket discards tokens.With Token Bucketa packet can only be transmitted if there are enough tokens to cover its length in bytes.
c. Leaky Bucket sends packets at an average rate. Token Bucket allows for large bursts to be sent faster by speeding up the output.
d. Token Bucket allows saving up tokens (permissions) to send large bursts. Leaky Bucket does not allow saving.

3. TCP Hybla: TCP Hybla aims to eliminate penalization of TCP connections that incorporate a high-latency terrestrial or satellite radio link, due to their longer round trip times. It stems from an analytical evaluation of the congestion window dynamics, which suggests the necessary modifications to remove the performance dependence on RTT.

4. TCP BIC: Binary Increase Congestion control is an implementation of TCP with an optimized congestion control algorithm for high speed networks with high latency.

5. TCP CUBIC: CUBIC is a less aggressive and more systematic derivative of BIC, in which the window is a cubic function of time since the last congestion event, with the inflection point set to the window prior to the event. Another major difference between CUBIC and standard TCP flavors is that it does not rely on the receipt of ACKs to increase the window size. CUBIC's window size is dependent only on the last congestion event.

6. Tahoe TCP: The Tahoe TCP implementation added a number of new algorithms and refinements to earlier implementations. The new algorithms include Slow-Start, Congestion Avoidance, and Fast Retransmit. The refinements include a modification to the round-trip time estimator used to set retransmission timeout values. The Reno TCP implementation retained the enhancements incorporated into Tahoe, but modified the Fast Retransmit operation to include Fast Recovery.The new algorithm prevents the communication path ("pipe") from going empty after Fast Retransmit.

7. New-Reno TCP:
We include New-Reno TCP in this paper to show how a simple change to TCP makes it possible to avoid some of the performance problems of Reno TCP without the addition of SACK.

8. Compound TCP: Compound TCP is a Microsoft implementation of TCP which maintains two different congestion windows simultaneously, with the goal of achieving good performance on LFNs while not impairing fairness. It has been widely deployed with Microsoft Windows Vista and Windows server 2008 and has been ported to older Microsoft Windows versions as well as Linux.

9. TCP Vegas: TCP Vegas is a TCP congestion control, or network congestion avoidance, algorithm that emphasizes packet delay, rather than packet loss, as a signal to help determine the rate at which to send packets. It was developed at the University of Arizona by Lawrence Brakmo and Larry L. Peterson.
The algorithm depends heavily on accurate calculation of the Base RTT value. If it is too small then throughput of the connection will be less than the bandwidth available while if the value is too large then it will overrun the connection. TCP Vegas is one of several varieties of TCP congestion avoidance algorithm. It is one of a series of efforts at TCP tuning that adapt congestion control and system behaviors to new challenges faced by increases in available bandwidth in Internet components on networks like Internet.

General Description
The Algorithm of TCP VEGAS are as follows:

a. Congestion detection during slow-start.
b. Congestion detection during congestion avoidance, more ag gressive fast retransmits mechanism.
c. Additional retransmissions for non-duplicate ACKs.
d. Prevention of multiple reductions of the congestion window in case of multiple segment loss.
e. Reduction of the congestion window by only 1/4 after a recovery (instead of halving it as in the case of TCP Reno).
f. Burst avoidance limits the number of segments that can be sent at once (that is, back-to-back) to three segments.
g. The congestion window is not increased if the sender is not able to keep up, that is, the difference between the size of the congestion window and the amount of outstanding data is larger than two maximum-sized segments.

Vegas - New congestion avoidance mechanism
Reno's congestion detection uses the loss of segments as a signal of congestion. It has no mechanism to detect the incipient stages of congestion - before losses occur - so they can be prevented. Reno is reactive, rather than proactive, in this respect. Reno needs to create losses to find the available bandwidth of the connection. Vegas is proactive algorithm TCP Vegas checks once every RTT whether it must modify its strategy for updating the congestion window. In slow-start, it checks whether it must give up the exponential opening of the congestion window and switch to congestion avoidance. When executing the check mentioned above, TCP Vegas resets baseRTT if only one segment has been transmitted during the last RTT. With the help of this reset, TCP Vegas may be able to cope with routing changes which increase

the minimum RTT.This algorithm is not in effect during slow-start.

The following are the other TCP congestion avoidance algorithms. They are;

FAST TCP H-TCP High Speed TCP HSTCP-LP TCP-Illinois TCP-LP	TCP SACK Scalable TCP TCP Veno Westwood YeAH-TCP TCP BaLDE

Conclusion:

As Congestion control is necessary to protect network from overload. Because too many sources sending too much data too fast for the network to handle. It leads to a major problem in networking. So hence,we have introduced several techniques for improving TCP, including a novel approach to congestion avoidance that tries to control the number of extra buffers the connection occupies in the network. Experiments on both the Internet and using a simulator show that Vegas achieves 40% to 70% better throughput, with one-half to one-fifth as many bytes being retransmitted as compared to the implementation of TCP in the Reno distribution of BSD Unix. In many respects, this solutions is still preliminary, as we have new algorithms are in the race to get the best of the best methods, by adopting the best features, we can get a good solution to the problem of congestion.

REFERENCES

1. L. Brakmo, S. O'Malley, and L. Peterson."TCP Vegas: New techniques for congestion detection and avoidance." SIGCOMM 94 2. Computer Networks: Andrew S. Tanenbaum 3. J. Mo, R. La, V. Anantharam, J. Walrand. "Analysis and Comparison of TCP Reno and Vegas."INFOCOM 99. 4. Performance Analysis of TCP Congestion Control Algorithms by Habibullah Jamal, Kiran Sultan 5. V. Jacobson, "Congestion avoidance and control,"the ACM SIGCOMM'88 6. ftp.cs.arizona.edu/xkernel/Papers/vegas.ps 7. www.wikipedia.com.