IPASJ International Journal of Computer Science (IIJCS)
Web Site: http://www.ipasj.org/IIJCS/IIJCS.htm
A Publisher for Research Motivation ....
Email: editoriijcs@ipasj.org
Volume 2, Issue 8, August 2014
ISSN 2321-5992

A Study on TCP Congestion Control Using RED Algorithm with SNR Ratio

T. Shanmuga Priya[1], Dr.N.R.Ananthanarayanan[2]

[1]Research Scholar, Department of Computer Science and Applications,
SCSVMV University, Enathur, Kanchipuram, Tamilnadu, India

[2]Associate Professor, Department of Computer Science and Applications,
SCSVMV University, Enathur, Kanchipuram, Tamilnadu, India

ABSTRACT

TCP (Transmission Control Protocol) is a protocol used along with the Internet Protocol (IP) to send data in the form of packets between computers over the Internet. While IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the packets that a message is divided into for efficient routing through the Internet. TCP considers all packet timeouts in wired networks as due to network congestion and not to bit errors. It therefore considers all packet losses as due to congestion and consequently reduces the burst of packet, diminishing at the same time the network throughput [1]. In Existing work the classical Slow-start algorithm along with reserved bits and SNR ratio has been developed for this problem. But Slow-start algorithm takes some time to receive the acknowledgement from the receiver. This paper proposes a new TCP RED congestion avoidance algorithm appropriate for wireless as well as wired networks and is capable of distinguishing congestion losses from error losses proactively. Proposed RED algorithm won't wait for the packet's acknowledgement. The proposed RED algorithm also based on using the reserved field of the TCP header to indicate whether the established connection is either wired or wireless link. Additionally, the proposed algorithm uses the SNR ratio to detect the reliability of the link proactively. Instead of the slow-start algorithm, proposed system uses the RED algorithm.

Keywords: TCP Congestion, RED, Reserved Field, SNR

1. INTRODUCTION

1.1 Congestion

Congestion occurs when a link or mode is carrying so much data. A situation is called congestion if performance degrades in a subnet because of too many data packets. The number of packets delivered is proportional to the number of packets send. But if traffic increases too much, routers are no longer able to handle all the traffic and packets will get lost. With further growing traffic this subnet will collapse and no more packets are delivered. Obviously, two naive solutions are possible: increase of resources or decrease of load [2]. Effects of congestion are queuing delay and blocking of some new connections. A router can only process one packet at a time. If packets arrive faster than the router, it puts them into queue until it can get around to transmitting them. 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 [3].

Nowadays, more and more applications require fast transfer of massive data over networks, and the emergence of high-speed networks provides an ideal solution to this challenge. Due to the limitations of the conservative congestion control algorithm, the standard TCP is no longer appropriate for high- speed networks to efficiently utilize the bandwidth resources [4]. Modern network uses congestion control and congestion avoidance techniques to avoid the congestion collapse.Congestion occurs when the number of packets being transmitted through the network approaches the packet handling capacity of the network. The Congestion effect will be that if too many packets are present in the subnet, a situation which causes performance degradation Similar to road congestion [5].

1.2 Congestion Solution

Congestion control and two basic approaches

- Open-loop: try to prevent congestion occurring by good design
- Closed-loop: monitor the system to detect congestion, pass this information to where action can be taken, and adjust system operation to correct the problem (detect feedback and correct) [7].

1.3 Congestion Control

Congestion control is intended to keep a fast sender from sending data into the network due to a lack of resources in the network (e.g., available link capacity, router buffers). Congestion control is concerned with the bottleneck routers in a packet switched network [6]. When one part of the subnet becomes overloaded then congestion results. Because routers

IPASJ International Journal of Computer Science (IIJCS)
Web Site: http://www.ipasj.org/IIJCS/IIJCS.htm
A Publisher for Research Motivation .... Email: editoriijcs@ipasj.org
Volume 2, Issue 8, August 2014 ISSN 2321-5992

are receiving packets faster than they can forward them, one of the two things must happen. The subnet must prevent additional packets from entering the congested region until those already present can be processed. The congested routers can discard queued packets to make room for those are arriving.

2. CONGESTION CONTROL ALGORITHMS

Congestion control is concerned with efficiently using a network at high load. Several techniques can be employed. These include the following:

- Warning bit
- Choke packets
- Load shedding
- Random Early Discard (RED)
- Traffic shaping

2.1 Warning Bit

A special bit in the packet header is set by the router to warn the source when congestion is detected. The bit is copied and piggy-blocked on the ACK and sent to the sender. The sender monitors the number of ACK packets it receives with the warning bit set and adjusts its transmission rate accordingly.

2.2 Choke Packets

The Choke Packets approach interprets the whole network as an active part of Flow Control. Therefore, each network actor has its own maximum of throughput rate and if it is exceeded so-called choke packets are send to the origin. These specially marked packets prevent further network nodes from generating equal Choke Packets and thus they prevent duplicate feedback. Furthermore, after a Choke Packet reaches the initial sender, it will throttle down its own output rate to an adequate level, within the time congestion is reported by further choke packets [2]

.2.3 Load Shedding

When buffers become full, routers simply discard packets. Which packet is chosen to be the victim depends on the application and on the error strategy used in the data link layer. For a file transfer it cannot discard older packets since this will cause a gap in the received data. For real time voice or video it is probably better to throw away old data and keep new packets. Get the applications to mark packets with discard priority. To implement a router the intelligence to decide which packet should be dropped applications must mark their packets in priority classes. This indication should be done in a very sensitive way, because it does not make any sense to mark all packets with a high priority [2].

2.4 Traffic Shaping

Traffic shaping is a generic term for a couple of algorithms avoiding congestion on sender’s side without feedback messages. Therefore, an essential decision - the data rate is negotiated either on connection set-up or is statically included in used implementations.

Two traffic shaping algorithms:

- Leaky bucket
- Token bucket

2.4.1 Leaky Bucket

The Leaky bucket Algorithm generates a constant output flow. The name describes to way of working: it works like a bucket with water and a leak on the bottom. How much water runs into the bucket does not matter. As long as there is any water left in the bucket it runs out at the same constant rate defined by the leak’s size. Obviously, if there is no water in the bucket there is no output. If the bucket is completely filled additional incoming water gets lost [2].

2.4.2 Token Bucket

The intention is to allow temporary high output bursts, if the origin normally does not generate huge traffic. One possible implementation uses credit points or tokens which are provided in a fixed time interval. These credit points can be accumulated in a limited number (= bucket size) in the bucket. In case of submitting data these credits have to be used from the bucket, i.e. one credit is consumed per data entity (e.g. one byte or one frame) that is injected into the network. If the amount of credit points is used up (the bucket is empty), the sender has to wait, until it gathers new tokens within the next time interval [2].

3. CONGESTION AVOIDANCE ALGORITHMS

- Slow-start algorithm
- Congestion avoidance
- Fast retransmit
- Fast recovery

3.1 Slow-Start Algorithm
Slow-start algorithm is used when congestion window cwnd < slow start threshold, while the congestion avoidance algorithm is used when cwnd> slow start threshold. When cwnd and slow start threshold are equal, the sender may use either slow start or congestion avoidance [12]. Slow start is part of the congestion control strategy used by TCP. Slow start is used in conjunction with other algorithms to avoid sending more data than the network capable of transmitting that is to avoid causing network congestion. Congestion window is one of the factors that determine the number of bytes that can be outstanding at any time. This is not be confused with TCP window size which is maintained by receiver. This is a means of stopping the link between two places from getting overloaded with too much traffic. The size of the window is calculated by estimating how much congestion there is between the two places. The sender maintains the congestion window [11].

3.2 Fast Retransmit
Fast retransmit Algorithm uses explicit feedback methods to avoid long timeout periods waiting for packet retransmitting in case of packet loss. Such problems are inherent in packet-switched data networks because every data packet can travel individually trough the rest of the network and can use special routes from the sender to the recipient. Consequently, the transmitted data packets will neither reach the recipient in accurate order nor complete continually.
Therefore, after detecting a missing packet the recipient sends duplicated ACK packets for the last correct received packet until the missing packet receives [2,11].

3.3 Fast Recovery
A special Congestion Avoidance Algorithm often combined with Fast Retransmit to restart transmission at a higher throughput rate than slow start does it is the fast recovery Algorithm. Fast Recovery starts when Fast Retransmit stops working. If no further duplicate ACK packets are received for Fast Retransmit Algorithm, the sender tries to return to normal sending state. But, instead of Slow Start Congestion Avoidance (additive increase) is used, because the returned duplicate ACK packets traveled successfully through the network. So, no congestion is present on this route at the present time and the sender can begin transmitting at a relatively high output rate specified by thresh [2].

4. RED ALGORITHM

The RED algorithm is a congestion avoidance technique used in communication networks to avoid network congestion. Compared to existing algorithms, RED monitors network traffic loads in an effort to anticipate and to avoid congestion at common network bottlenecks, where the system triggers before any congestion actually occurs [8]. To reduce their transmission rate before queue overflow and packet loss occur using active queue management [9]. RED can decide to drop each arriving packet with come drop probability whenever the queue length exceeds some drop level. This idea is called early random drop. The RED algorithm defines the details of how to monitor the queue length and when to drop a packet. First, RED computes an average queue length using weighted running average similar to the one used in the original TCP timeout computation. If the average value exceeds threshold value then it won't allow send the packet [10].

5. RELATED WORK

TCP suffers from performance degradation over error-prone wireless links as it has no mechanism to differentiate error losses from congestion losses. It therefore considers all packet losses as due to congestion and consequently reduces the burst of packet, diminishing at the same time the network throughput. This paper proposes a new TCP congestion control scheme appropriate for wireless as well as wired networks and is capable of distinguishing congestion losses from error losses. The proposed scheme is based on using the reserved field of the TCP header to indicate whether the established connection is over a wired or a wireless link. Additionally, the proposed scheme leverages the SNR ratio to detect the reliability of the link and decide whether to reduce packet burst or retransmit a timed-out packet [1].

6. PROPOSED WORK

Network congestion occurs when a link or node is carrying so much data. Wired network and Wireless network both affected by the congestion. TCP suffers to distinguish the packet losses such as congestion losses from error losses. The existing scheme is based on using the reserved field of the TCP header to indicate whether the established connection is over a wired or a wireless link. Also it uses the SNR ratio to find the reliability and the noise. The proposed scheme uses the SNR ratio and reserved field with the RED algorithm instead of slow start algorithm. RED algorithm is the proactive algorithm. So the algorithm can detect the congestion before the packet send to the destination. RED algorithm won't wait for the packet's acknowledgement. But classical Slow-start algorithm waits for few minutes to receive the acknowledgement. After receiving the acknowledgement, it increases the congestion window size. If it not receives the acknowledgement, it will reduce the size of the window congestion window (cwnd). RED algorithm

IPASJ International Journal of Computer Science (IIJCS)
Web Site: http://www.ipasj.org/IIJCS/IIJCS.htm
A Publisher for Research Motivation ....
Email: editoriijcs@ipasj.org
Volume 2, Issue 8, August 2014
ISSN 2321-5992

computes the average queue length average each time a packet arrives. It checks the average with the threshold value then it will send the packet. This Proposed RED algorithm is used for congestion avoidance and also quickly distinguishes the congestion losses and error losses of packets.

6.1 Signal-to-noise ratio:

Signal-to-noise ratio (SNR) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. It is defined as the ratio of signal power to the noise power, often expressed in decibels.

$$SNR = \left(\frac{P_{signal}}{P_{noise}}\right) \qquad (1)$$

Where P is average power. Both signal and noise power must be measured at the same and equivalent points in a system, and within the same system bandwidth [1, 13]. Often SNR is expressed in decibel units according to the following equation

$$SNR_{dB} = 10 log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) = P_{signal,dB} - P_{noise,dB} \qquad (2)$$

6.2 Proposed Algorithm

The proposed RED algorithm when operating in wireless mode exploits the SNR ratio of the communication line to decide whether a timed-out packet was due to congestion or error loss. When a TCP connection first begins, the alternative RED algorithm computes the average of the queue. Then it checks the average value with the threshold value. If the average is less than the threshold value, the packet will be queued. If the average is greater than the threshold value the algorithm checks the reserved bits which indicates the connection is wired or wireless [1]. The first bit of the reserved field is set according to the type of the link, i.e., b=0 for wired and b=1 for wireless connection. If it is equal to 0 (wired link), then timeout is considered to be due to congestion. So the packet will be discarded. In contrast, if the reserved bit *b* is equal to 1 (wireless link), the SNR ratio of the connection is checked. In case it is within a high range, i.e., greater than 5dB (SNR>5dB), then timeout is considered to be due to congestion and the packet will be dropped. However, if SNR ratio is within a low range, i.e., less than 5dB (SNR<5dB), then timeout is considered to be due to error and the timed-out packet is retransmitted to the receiver. The flowchart of the proposed **RED algorithm is depicted in Figure 6.2; while, its pseudo-code is outlined as follows:**

1. Computes the queue average value
2. Check the average avg with threshold value
3. If avg<threshold packet will be queued for sending to receiver.
4. If avg>threshold check the reserved bit b
5. If the reserved bit value b is 0 the connection is consider to be wired and the timeout due to congestion so the packet is discarded.
6. If the reserved bit value b is 1 the connection is consider to be wireless and the SNR ratio will be checked.
   1. If the SNR>5dB then the timeout due to congestion and the packet will be dropped.
   2. If the SNR<5dB then the timeout due to error and the packet will retransmit to the receiver.

IPASJ International Journal of Computer Science (IIJCS)
Web Site: http://www.ipasj.org/IIJCS/IIJCS.htm
A Publisher for Research Motivation ....
Email: editoriijcs@ipasj.org
Volume 2, Issue 8, August 2014
ISSN 2321-5992

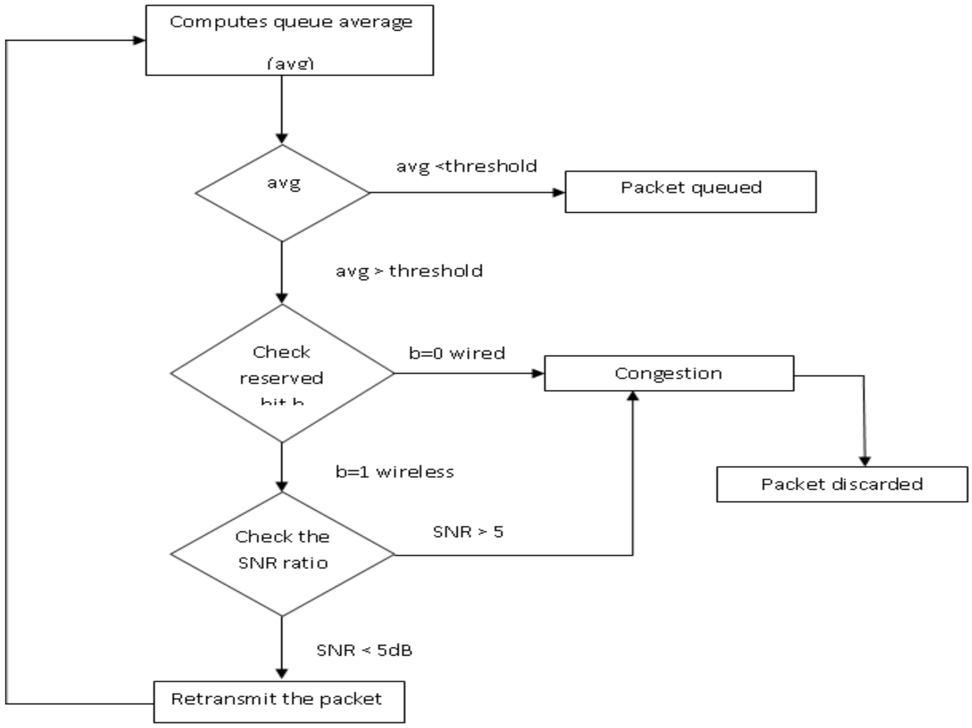

Figure 1 Flowchart of the Proposed Congestion RED Algorithm

6.3 Existing RED algorithm

Initialization
avg ← 0
count ← -1
for each packet arrival
calculate new avg of queue size
if the queue is nonempty
$avg \leftarrow (1 - w_q)\, avg + w_q\, q$
else
$m \leftarrow f\,(time - q\_time)$
$avg \leftarrow (1 - w_q)^m\, avg$
if $min_{th} <= avg < max_{th}$
increment count
calculate probability p_a :
$p_b \leftarrow max_p\, (avg - min_{th}) / (max_{th} - min_{th})$
$p_a \leftarrow p_b / (1 - count.p_b)$
with probability p_a :
mark the arriving packet
count ← 0
else if $max_{th} <= avg$ [15]

6.5 Enhancement of RED algorithm

mark the arriving packet
count ← 0
else b=1 then wired
if SNR>5dB
mark the arriving packet
count ← 0
else SNR<5dB
mark the arriving packet
count ← 0
retransmit the packet
else count ← -1
when queue becomes empty

$q\_time \leftarrow time$

Saved variables

avg: average queue size

q_time: start of the queue idle time

count: packets since last marked packet

Fixed parameters

w_q: queue weight

min_{th}: minimum threshold for queue

max_{th}: maximum threshold for queue

max_p: maximum value for p_b

Others

p_a : current packet-marking probability

q: current queue size

time: current time

7. SIMULATION WITH NS2

NS2 uses two languages, because it has two things to do. On one hand, detailed simulations of protocols requires a system programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks, run-time speed is important. C++ is slow to change, but its speed makes it suitable for protocol implementation [13]. Here the theory analysis work is presented the simulation work with other features can done in future.

8. CONCLUSION

This paper presented a proposed RED algorithm which solves the congestion avoidance problem. Its aim is to allow the TCP protocol to distinguish between transmission timeouts due to congestion and those due to error proactively. The goal of this project is to avoid the congestion proactively. In existing work classical slow-start algorithm is used with SNR ratio and the reserved bits of TCP. Slow-start algorithm waits for some time to receive the acknowledgement from the receiver. In this proposed RED algorithm shouldn't wait for any acknowledgement. It will avoid the congestion proactively.

9. FUTURE WORK

For future work the proposed RED algorithm can be run in C++ code in ns2. Because TCL code is not efficient for this proposed algorithm. CRC and HEC metrics are to be added as additional parameters to the proposed scheme so that it can better determine and predict the cause of transmission timeouts in wireless networks. Instead of this RED algorithm the other congestion control and avoidance algorithms can be added with the SNR ratio and the reserved bits. This RED algorithm drops the packet when the congestion is occurring. So in future work the algorithm can be enhanced with reduced dropping packets counts with the proved simulation work.

REFERENCES

[1] Youssef Bassil "TCP Congestion Control Scheme for Wireless Networks based on TCP Reserved Field and SNR Ratio", IJRRIS, June 2012.

[2] Alexander K¨uchler, Matthieu-Patrick Schapranow "Congestion Control", COMMUNICATION NETWORKS, winter semester 2004/2005 Seminar.

[3] V.Jacoban "Congestion Avoidance and Control", Proc. SIGCOMM '88, Vo1 18 No. 4, August 1988

[4] Renu Dangi and Neeraj Shukla" A New Congestion Control Algorithm for High Speed Networks", International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 1. Dr. Z. Huang "Congestion Control", TELE202 Lecture 8

[5] Peterson, Davie and Morgan Kaufmann "Congestion Control and Resource Allocation", Computer Networks A Systems Approach, 2007.

[6] S Chen "Congestion Control Overview", ELEC3030 Computer Networks.

[7] Sridhar Madipelli, David Raj Gillella and Sudhakar Devaraya "The RED Algorithm – Averaged Queue Weight Modeling for Non Linear Traffic",

[8] Blekinge Institute of Technology, November 2009.
[9] Brijendra singh "Data communication and computer networks", 2nd Edition, PHI publications.
[10] Larry L. Peterson and Bruce S. Davie "Computer Networks- A Systems approach", Fifth Edition.
[11] Lydia Parziale, David T. Britt, Chuck Davis, Jason Forrester, Wei Liu, Carolyn Matthews and Nicolas Rosselot "Redbooks: TCP/IP Tutorial and Technical Overview", www.ibm.com/redbooks, Eighth Edition, December 2006.
[12] R.Boder and C.G. Lee, "Real-time guarantee of a periodic packets in single-hope ad hoc wireless networks", IEEE, 2005.
[13] SNR Ratio- www.wikipedia.org[online]
[14] Arijit Ganguly and Pasi Lassila "A study of TCP-RED congestion control using ns2", July, 2001
[15] Sally Floyd and Van Jacobson "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM TRANSACTIONS ON NETWORKING. VOL I. NO 1. August 1993