Flow Control Mechanisms - TRUMP

A.Caroline Mary
Research Scholar, Hindustan College of Arts and Science College
Asst Professor, Sri Krishna Arts and Science College
Coimbatore, India
marypsgtech@gmail.com,carolinemarya@skasc.ac.in

Dr. A.V Senthil Kumar
Professor, Department of MCA,
Hindustan College of Arts and Science,
Coimbatore - 28
avsenthilkumar@yahoo.com

***Abstract*—**

Networks - Collection of Nodes form a Network. There is a necessity to share information in a Network. Packet loss and delay should be addressed on scenarios like this. No Packet should wait for a Longer time. To solve this n number of Scheduling algorithms are available in the market. We should also make sure that the Packet is not lost while transmitting, because of the Congestion or traffic in the network. Flow control Mechanisms are available to avoid traffic in the network. Let us look in to the mechanisms to control flow in the network and add-ons to it which can further enhance the performance of the Network . Criteria like Packet Loss, Re Transmission, Window Size, Duplicate Ack and selective ack are considered to enable a Traffic less transfer of data in the Network. Fast sender and Slow Receiver issues are addressed and discussed in a effective way. Issues like Bufferbloat should also be treated

Keywords-Flow control, Network, Packet, Scheduling

Introduction

In exchange of data, n number of things are to be considered. Eg. Which Packet should be scheduled 1st. What is the time for the Packet to be placed in the Link. What is the time for the Packet to reach buffer of Destination and finally from buffer to the destination and so on. The problem is the difference between the sender and the receiver.

Sender may be capable of Placing more number of packets in the link, but the capacity of destination might be very less. In turn the Packet will not be processed at the destination , which leads to loss of Packets. To solve this, according to the Receivers capacity data should be sent to the destination. TCP uses Flow control Mechanism - Sliding window Protocol to solve the Congestion issue.

II Flow Control Mechanisms

TCP uses sliding window Protocol to control Traffic in the network. The destination sends the sender the required packet that can be sent through the window size. According to the window size, the sender sends the data. Thereby Collision or traffic does not occur in the network.

Example a window size of 3. The sender fills the Packet accordingly. Slow start Phase options are enabled. Thereby no. of Packets are increased 1 by 1 and checked whether packet loss occurs.

During the retransmission of Packet loss, the window size is extended , thereby the faster delivery of Packets.

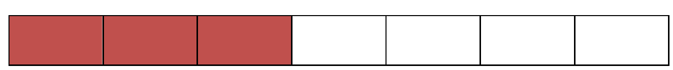

fig 1 . Sliding window

To the receiver's buffer only 4 Packets are sent as the window size is 4. As of Stop and wait protocol, for each and every Packet, ACK will be sent back. Which wastes the link utilization. Whereas in the case of Sliding window , ack is not sent for every packet. Eg. ACK 3 will mean that Packet 0,1,2 have successfully received

.

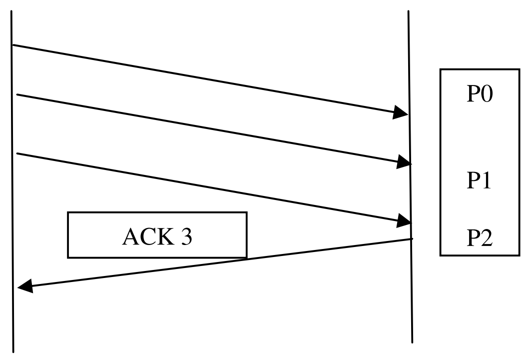

Fig 2. Sliding window withACK

III TCP RENO

A smaller window size, depicts that only few packets can be sent. A Larger window size is expected to make sure that large amount of Packets are sent. On having a smaller window size we can be sure that there will not be any congestion in the Network.

Eg. Window size of 1

But if we are sure that there will not be any congestion in the Network, we have to increase the Window size. The Thing here is when to increase or decrease the window size and by what extent. If there is a probability for Packet loss, the window size should be decreased gradually. Whereas in the case of TCP RENO, only on Packet loss the window size is decreased.

Servers generally allocate equal bandwidth to all its client, thereby fairness of resource allocation is available.

III Flow control in TCP

In the case of TCP Flow control, there is a need for us to mention the window size of receiver.

The window size is not static in nature. It can be changed dynamically according to the scenario.

Initially a small window size can be mentioned. And if there are no loss in Packets, the window size can be increased gradually.

Duplicated Acknowledgement

If there is a loss of Packet in the Network. It can be found out only after the Time out. But there is a delay in this. To solve this we have a Duplicate Acknowledgement . In this case, if there is a loss in Packet, thrice the ACK for the Previous packet is sent. Which instructs the sender about the loss of the Packet. And sender further sends the Packet again.

Fast retransmission can be enabled for the lost packet, as it might be already late to send the Packet again. Faster retransmission can be enabled by increasing the window size.

Eg. Packets 0,1,2,3,4,5 are sent in the Network. On loss of Packet '2', False ACK 2 will be sent thrice.

synchronized-loss hypothesis

Two connections will experience the same number of losses as a long-term *average*, it is a much stronger statement to suppose that all loss events are shared by both connections

IV Proposed Flow control in TCP - TRUMP

In general, the sender after sending the Packets in the Link, it must wait for the ACK from the receiver. Till it receives an ACK, it just waits. Though this seems to be a disadvantage. This gives sender sometime to gather the Packets that it has to send. The reason is even if we have to send only 1 bit of Data, the entire TCP packet should be sent in the network, which has source address, destination address and so on.

Sequence number is also available, which helps us in reordering the Packets. There is a field ' Window' which guides us in knowing the window size. And finally the literal data to be sent in the network is also added .

Checksum field, will help us in Error control. Data that is sent from the sender, should reach as the same in to the Receiver. To check for any deviations, checksum can be helpful.

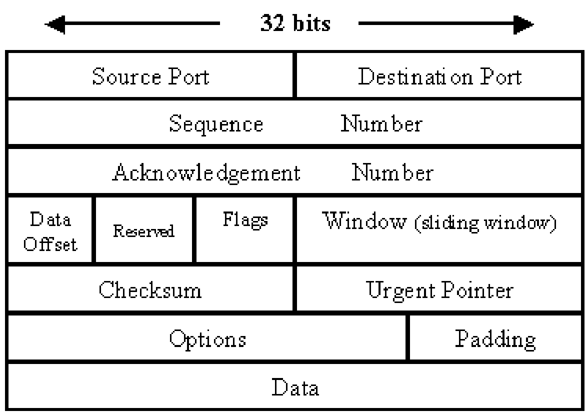

Fig 3. TCT Packet

Though we are only sending a small data, the entire packet should be sent. So it's better to have enough data in the Packet that is to be sent.

Congestion control is the major task here. As each packet is important and loss of packets and retransmission of the Packet wastes the resources.

Because of timeout , if the Packet is lost. The sliding window pipe will be drained. Sliding window should start from scratch.

Additive increase and Multiplicative Decrease

When there is no congestion in the Network Additive increase in Window size can be done.

eg. Window size = window size+ 1

On congestion in the Network, the issue has to be addressed immediately. And the window size should be decreased .

Eg. Window size = Window Size / 2

INCREASE	**DECREASE**
12	6
13	3
14	1

Fig 4. Window Size

Bufferbloat

To make sure that large amount of data can be sent from sender to receiver, we prefer a Large buffer.

But as the buffer size increases, the packets are made to wait in the queue for a long time. If the Packets of Large flow are made to wait in the Queue. The small flows will have to suffer and wait for a very long time in the Queue.

STEPS IN Proposed System - TRUMP

1. Start with window size of 1
2. On acknowledgement from the receiver, the receiver also sends the preferred Window Size
3. According to Window size, further send the Packets to receiver.
4. On loss of Packets don't wait for timeout or 3 False Ack
5. Work on Selective Acknowledgement. Eg. Packet 1000 to 1050 is received except 1003
6. On Congestion in Network, divide the Window size by 2.
7. On less traffic, increase the Window size by 1
8. Give Fairness to all connections
9. Divide the receiver buffer in to 2
10. 1 to accommodate large Flows, and 1 to accommodate Small Flows
11. On Packet retransmission, increase the size of the Window.

ALGORITHM

Window Size = 1

if(congestion)

{

Window_size = Window _size /2;

```
}
else
{  Window _size = Window_size +1; }
if( small_Flow)
{
   small_buffer = Packet + 1;
}
else
Big_buffer = Packet +1;
if( Packet_Loss)
{
  Selective Acknowledgement
  Window_size = Window_size+1; }
if(More_connections)
{
   Fairness in no. of Packets;
}
```

CONCLUSION

Flow control mechanisms are available in TCP. But on Packet Loss 3 False ack is preferred sometimes, which does not utilize the bandwidth effectively. Selective Acknowledgement method can be preferred. Though the Larger buffer is preferred in general to send data, there is a possibility of Buffer Bloat. Which causes the smaller flows to suffer. An additional queue can be used to enhance the processing of small Flows.

Equal Fairness is tried to maintain. Faster Sender and slow receiver issues should be further considered to utilize the bandwidth effectively in the networks.

REFERENCES

[1] Hiroyuki Koga1, Katsuyoshi Iida1, and Yuji Oie2, Receiver-Based Flow Control Mechanism with Interlayer Collaboration for Real-Time Communication Quality in W-CDMA Networks. pp. 286–300, 2004

[2] **A. Caroline Mary, Dr. A. V. Senthil Kumar** Effective scheduling in Networks - TRUMP . IJSRNSC. Volume 5, Issue 6

[3]. Zhenzhen Jiao, Wei Gong, Cheng Li, , Baoxian Zhang. Virtual Gradient based Back-Pressure Scheduling in Wireless Multi-Hop Networks. . *IEEE ICC 2015 - Mobile and Wireless Networking Symposium*. PP. 3281 - 3286

[4]. Suhas Bansode, Santosh Sambare Performance Evaluation of Dynamic Multilevel Priority (DMP) Packet Scheduling Method for Wireless Sensor Networks (WSNs). *International Conference on Pervasive Computing (ICPC)* 2015.

Biographies

A. Caroline Mary
PhD Student, Hindustan College of Arts and Science, Cbe.*Assistant Professor, Department of ComputerTechnology, Sri Krishna Arts and Science College, Tamilnadu, India*
Area of Interest : Network Security, Operating System
Email: marypsgtech@gmail.com

Dr. A. V. Senthil Kumar
Professor, Department of MCA, Hindustan College of Arts and Science, Coimbatore .
avsenthilkumar@yahoo.com