



RESEARCH ARTICLE

PACKET HANDLING SCHEDULING IN MULTIPLE ROUTING CONFIGURATIONS FOR FAST IP NETWORK RECOVERY

H.V. Laxman Vijay¹, K. Sudheer Kumar², R.V. Krishnaiah³

¹Department of computer science Engineering, DRK College of Engineering & Technology, India

²Department of computer science Engineering, DRK College of Engineering & Technology, India

³Department of computer science Engineering, DRK College of Engineering & Technology, India

¹ hvlaxmanvijay@gmail.com; ² sudheerkumar19@gmail.com; ³ r.v.krishnaiah@gmail.com

Abstract— As a promising approach to improve network reliability, Proactive Failure Recovery (PFR)^[1] re-routes failure affected traffic to backup paths without waiting for the completion of IP routing convergence. However, the failure affected traffic may cause congestion. If a proper scheduling policing mechanism is not used when a router redirect the traffic to alternative paths an overflow of packets may occur and degrade the performance. Therefore for reliability and from congestion free here we proposed a new scheme called WFQ^[2] mechanism which regulates the packets using different classes and makes the packet transmission without any loss in a packet.

Key Terms: - Computer network reliability; Class-Based Queuing(CBQ); failure recovery; FIFO; Fair Queuing(FQ); protection; Quality of service(QoS); routing; Traffic Engineering; Weighted Fair Queuing(WFQ)

I. INTRODUCTION

The biggest problems in a network are related to the allocation of network resources, as buffers and link bandwidth, to different users when node or link failures. A limited amount of resources has to be shared among many different competing traffic flows in an efficient way in order to maximize the performance and the use of the network resources. The behavior of routers in terms of packet handling can be controlled to achieve different kind of services.

It is a primary design goal of the Internet to continue to function despite of the node or link failures. At the IP level, when routers or links fail, the network should still be able to deliver the packets as long as alternative paths exist. In current routing protocols, such as OSPF (Open Shortest Path First) and IS-IS [5] (Intermediate System to Intermediate System), routers are informed about network topology changes by update messages, and then re-calculate their routing paths accordingly. However, this approach incurs convergence delay, as it takes considerable long time for the update messages to propagate throughout the network and for the routers to re-calculate routing path.

Therefore a different approach for handling physical failures is proactive failure recovery(PFR) in which routers compute and store backup paths for potential failures beforehand and once a failure is detected, a router will redirect traffic to backup paths right away instead of waiting for the completion of network wide routing convergence.

But here routers redirect traffic affected by failures to alternative paths without considering the load balancing. The diverted traffic will easily cause uneven traffic reassignments, which either congests on already heavily loaded links. Especially if other links are congested due to the diverted traffic, it defeats the goal of minimizing the impacts of failures to the application performance.

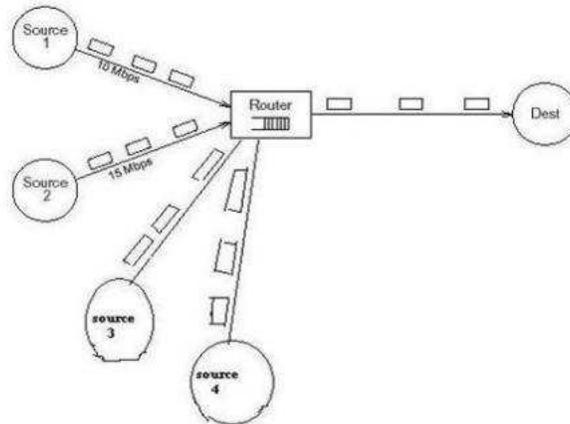


Fig 1: small congested network

II. CONGESTION

When too many packets rushing to a node or a part of network, the network performance degrades, and this situation is called as a congestion. When the number of packets dumped into the subnet and as traffic increases the network is no longer able to cope and design losing packets at very high traffic, performance collapses completely and almost no packets are delivered.

III. EFFECTS ON CONGESTION

The effect on congestion on throughput of a network is shown in the fig below:-

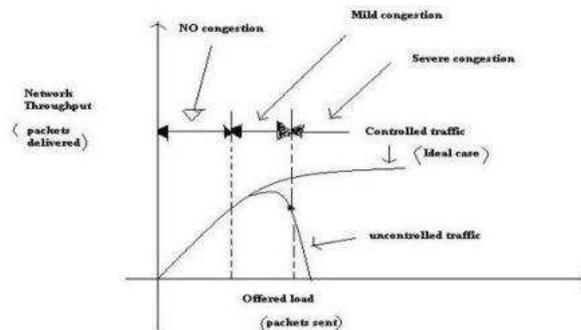


Fig 2 : Effect of congestion

Ideally as offered load increases, throughput also increases. Practically throughput drops with increase in offered load because the buffers at each node are full and starts discarding packets. Therefore source station must retransmit the discarded packets in addition with the new packets. Under these circumstances, the network utilization and hence performance falls off.

IV. CONGESTION CONTROL

Congestion control is a process of maintaining the number of packets in a network below a certain level at which performance falls off. Congestion control makes sure that subnet is able to carry the offered traffic. Hence to avoid congestion when routers redirect the packets to other links existing scheduling mechanism fails to work

effectively, so therefore in this paper we present a new scheduling scheme called weighted fair queuing(WFQ)[2][3][5] minimizing the effect of improper load balancing mechanisms.

V. WFQ SCHEDULING DISCIPLINE OVERVIEW

Packets from different networks are to be multiplexed and queued in buffer for transmission on a link. The method of selecting queued packets for transmission is called as link scheduling discipline. It plays an important role in providing better quality of service (QOS) [6].

In a network, packets are accumulated and queued into memory buffers of routers and switches. Packets may arrive at routers in bursts from multiple devices and a node may sometimes receive more packets than it can process. Buffers hold packets until the router can handle them. If the router is overloaded, buffers fill up and overflow. The most common methods applied in existing system are “tail drop or fifo or generalized processor sharing (gps)[7]” schemes as they work fast. But they are not that fair manner why because in tail-drop method it consists of dropping the new incoming packets that does not allow handling packets of different classes in different way and fifo & gps does not provide better bandwidth services. Therefore WFQ scheduling mechanism was presented for best result and for fair manner.

VI. WEIGHTED FAIR QUEUEING

In WFQ the arriving packets are classified and queued in several classes, scheduler serves all classes of queues in circular manner. First class-1 is served then class-2 is served then class-3 is served and the service pattern moves on to the next class. And thus the service pattern is repeated in the figure below:-

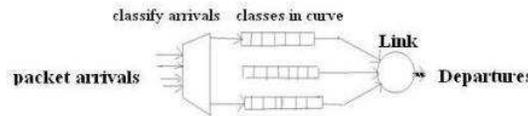


Fig 3: weighted fair queuing

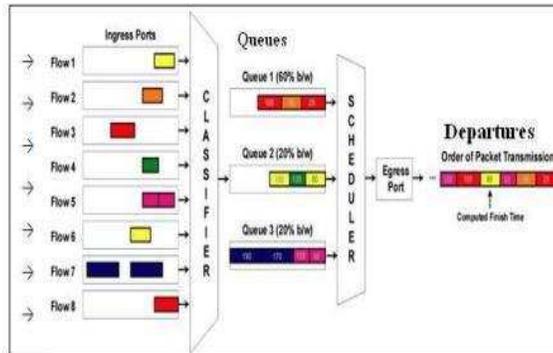


Figure 4: Weighted Fair Queuing Techniques (WFQT)

Weighted fair queuing (WFQ) is a packet scheduling technique allowing guaranteed bandwidth services. The purpose of WFQ is to let several sessions share the same link. The WFQ implementation primarily consists of three routines: get session identity, enqueue and dequeue. The data structures are one FIFO queue per session in form of a circular buffer as well as header information for each queue. There is also a sorted array with session identities. The get session identity function performs a binary search in the array. Only the virtual finish time (f) of the first packet in each queue is of interest at any given time. So, instead of storing F for each packet an F for the first packet in each queue is stored in the queue header. When a packet is dequeued the corresponding F is updated accordingly. Time is increased in steps each time a packet is dequeued. The arrival time of packets are set to the depart time of the packet currently being outputted, which is a deviation from true WFQ. Virtual time is updated every time a packet is dequeued since it is then time is altered and since arrival time of packets are set to these times too. Virtual time progresses at different speeds depending on the amount of active bandwidth, therefore we need to find out at which times sessions become inactive.

The virtual time a session becomes inactive is the F of the last packet in the session, the session virtual finish time (sf). The smallest SF. The smallest SF among active sessions is found and the corresponding time is calculated. If this time is less than the current time then this session is deactivated and the active bandwidth is decreased accordingly. This is done iteratively until the corresponding time is more than the current time. When this happens the session in question is not inactive yet and instead the virtual time that corresponds to the current time is calculated.

VII. ALGORITHM

```

enqueue(packet , i )
1  if not active( i )
2    activate( i )
3    active_r += r( i )
4  if queue( i ) is emptyt
5    F( i ) = SF( i ) = max(F( i ) , v( t )) + L / weight
6  else
7    SF ( i ) += L / weight
8  put (packet , queue( i ))
    
```

dequeue()

```

1  i = min(active queues F( i ))
2  packet = get (queue( i ))
3  t += L / r
4  if active ( i )
5    F( i ) += Lnext / r( i )
6  for ever
7    j = min ( active queues SF( j ))
8    tmp_t = prev_t + (SF ( j ) - v( t )) * active_r / r
9    if tmp_t > t
10     V( t ) += (t - prev_t) * r / active_r
11     Prev_t = t
12     return packet
13     Prev_t = tmp_t
14     V ( t ) = SF ( j )
15     deactivate( j )
16     active_r - = r ( j )
    
```

VIII. IMPLEMENTATION

The implementation of WFQ for network congestion free mainly uses 2- modules.

1. Client Module 2. Server Module 3. Router Module.

1. Client

This module is used to send the data to server through routers. It will provide user friendly interface to send the data to the required destination.

2. Server

It will receive the data send by the client which came from the active router. It can have any no. of clients.

3. Routers

These are placed in between server and client to transfer the data. Whenever client sends the data to the server it will pass through any router.

IX. CONCLUSION

We have presented weighted fair queuing (WFQ) [4][5] scheduling mechanism to achieve congestion free control. WFQ is working based on that the arriving packets are classified and queued in several classes; scheduler serves all classes of queues in circular manner.

REFERENCES

- [1] Tommy Andre, Skancke nyquist, "Evaluating local proactive recovery schemes for ip networks" Master thesis.
- [2] Ian Marsh "Implementation of weighted fair queuing" October 21,1997.
- [3] McKe90] P. McKenney "Stochastic Fairness Queuing", IEEE INFOCOMM'90 Proceedings, San Francisco, 1990.
- [4] [CBQ] <http://www.icir.org/floyd/cbq.html>. References on CBQ (Class-Based Queuing).
- [5] D. Oran, "OSI IS-IS intra-domain routing protocol," IETF Request for Comments 1142, February 1990.
- [6] Huston00] Huston, Geoff, "Internet Performance Survival Guide: QoS strategies for multiservice networks". John Wiley & Sons, February 2000.
- [7] A. parekh. "A Generalized processor sharing Approach to flow control in Integrated services networks". phd dissertation, Massachusetts Institute of Technology. February 1992.
- [8] A. Demers, S.Keshav, and S.Shenker. "Analysis and simulation of a fair queueing algorithm". In journal of Internetworking Research and Experience, pages 3-26, October 1990. Also in proceedings of ACM SIGCOMM89, pp 3-12.

AUTHORS BIBLIOGRAPHIES



H.V. Laxman Vijay is a student of DRK College of Engineering & Technology, Ranga Reddy, and Andhra Pradesh, India. He has received B.TECH degree in computer science Engineering & currently pursuing M.TECH Degree. His main research interest include on networks



K. Sudheer Kumar is working as an Asst Prof at DRK College of Engineering & Technology, and Andhra Pradesh, India. He has received M.TECH Degree in computer science engineering.



R.V. Krishnaiah is working as principal at DRK Institute of Science & Technology, Ranga Reddy, Andhra Pradesh, India. He received M.TECH Degree (EIE & CSE) & PH.D. His main research interest includes data mining, software engineering.