

Survey of Bandwidth Management Techniques

N.Snehalatha, S.Angeline Julia, Paul Rodrigues

Abstract:- Today in the modern communication world, the traffic that exists in the internet is becoming more and more abnormal. This was mainly due to increase in the number of users day by day which results in bandwidth congestion, poor response time for end users. The most efficient solution to this problem to manage and allocate the existing bandwidth almost equally using suitable queuing disciplines and filters that exist as quality of service. It is a full featured technology which may reduce the cost and improve the network performance. This study comprehensively surveys various bandwidth management techniques. This paper gives the brief overview of bandwidth management system and bandwidth management techniques

KeyWords: Bandwidth, techniques, paramete

I. INTRODUCTION

Bandwidth can be defined as the rate of data transmission. It is measured in bits per second or hertz. The process by which bandwidth is controlled and measured is called bandwidth management. A bandwidth manager is simply a gateway to better management of your network thus optimizing the Quality of Service for the network, while a shared internet connection is used. It should be capable of reducing the risk of lost information due to critical conditions.

The responsibilities of the bandwidth manager are :

1. Reducing bandwidth requirements
2. Tracking bandwidth uses
3. Ensuring security
4. Protecting business by disabling access of blacklisted sites
5. Ensuring suitable allocation of resources among all workstations

A network bandwidth manager maintains optimal accessibility and usage of critical applications and information related to business. It is very much important to detect the performance of the network in any organization. If the available bandwidth is increased the speed enhancements for file based transmissions show a more effective performance. In some cases of streaming application the necessary bandwidth may not be achieved.

- Main objectives of Bandwidth management is improve the QOS of Networks

1.1 Three Policy Type

Most bandwidth managers are policy based Gateways, in which a network administrator can define policy rules to solve the above mentioned problems. Each policy rule contains the *condition* and the *action* field's to define the specific actions for the specific conditions. The condition defines the packet-matching criteria to group certain

Manuscript received on July, 2013.

N.Snehalatha, Department of CSE, SRM University, India.

S.Angeline Julia Assist Prof, Department of CSE, SRM University, India.

Dr. Paul Rodrigues, Dean Research School of Computing Velammal Eng.College, India.

Subnet/application/protocol into a bandwidth class. The action defines the parameters for a bandwidth class, such as 100 kb/s. The complexity of a policy rule ranges widely. For example, a 125 kb/s access link may be partitioned into a 90 kb/s VoIP class and a 35 kb/s FTP class (class-based bandwidth limitation). If there is no voice call, FTP sessions can occupy the entire 125 kb/s link. Whenever a 30 kb/s VoIP session starts, the bandwidth manager allocates 30 kb/s for the VoIP class until the 90 kb/s are used up by the three 30 kb/s voice calls. This is defined as *inter-class bandwidth borrowing* where FTP can borrow the unused bandwidth from the VoIP class. suppose three FTP sessions are mixed in the 35 kb/s class. Each session can be isolated to obtain a guaranteed 11.6 kb/s (*session bandwidth guarantee*).

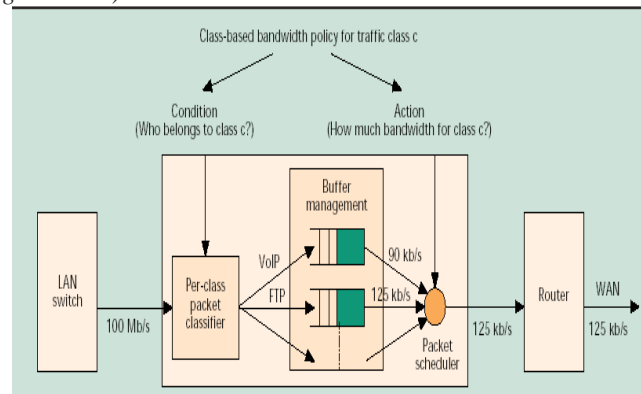


Fig-1. Building Block of Bandwidth Management System

II. INTERNAL STRUCTURES

The internal structure of a bandwidth manager is as follows. In Fig. 1, a class-based bandwidth policy matches packets of Certain protocols/subnets into a per-class queue according to the specified criteria set in the packet classifier. The buffer manager then optimizes the queuing behaviors in the per-class queue. Finally, the traffic in the class is scheduled out at its corresponding specified bandwidth. To correctly set up a Bandwidth management device involves the following steps:

Step 1. Defining the WAN-Link Bandwidth: In Fig. 1, the subscribed WAN link is 125 kb/s. If the traffic from the bandwidth manager to the router is more than 125 kb/s, queuing must occur at the router and may cause buffer overflows. Thus the first step to configure the bandwidth manager is to limit the traffic pumping into the router. In Fig. 2, a configured WAN-link bandwidth of 125 kb/s solves the problem.

Step 2. Partitioning the WAN Link: In Fig. 1, after the configuration in Step 1, the 125 kb/s is partitioned into the 90 kb/s and the 35 kb/s classes. The next step then defines who belongs to which class.

Step 3. Classification by the Packet Classifier: It is only when the passing traffic is correctly identified that the policy Can be accurately enforced. In Fig. 1 packets are classified into their corresponding queues.

Step 4. Optimization of Buffer Management: After packets are classified into the queues, the queuing behaviors dominate the effectiveness of the policy enforcement. In Fig. 1, buffer management techniques are employed to optimize the queuing behaviors. Based on the input traffic types, the techniques used can be categorized into.

Step 5. Scheduling by the Packet Scheduler: The packets in the queues (Fig.1) are waiting to be scheduled out onto the Wire according to their specified bandwidth parameters.

III. OVERVIEW OF BANDWIDTH MANAGEMENT TECHNIQUES

In the section we discuss some bandwidth management techniques can be used to improve the most straight forward method for managing the bandwidth as already shown in fig.1, is queuing First, it requires a *packet classifier* to classify packets into their corresponding queues according to user-defined policy rules. Secondly, queuing requires a *packet scheduling* algorithm that can decide from which queue comes the next packet for transmission.

3.1 SCHEDULING

Packets from different flows arrive at a switch (or) routes for processing. Scheduling is a basic component to control the bandwidth several scheduling techniques are designed to improve the quality of service.

3.1.1 FIFO QUEUING

In **first-first-out (FIFO) queuing** is first processed is a new packet arrives they are added at the end of the packets wait in the queue until the node is ready to process.

3.2 Traffic shaping could be incorporated to optimize some performance metric, such as buffer requirements & queue delay.

3.3 Buffer management scheme may be employed within some queues to achieve specific purposes, such as fairness among connections.

3.1.2 PRIORITY QUEUING

In priority queuing, packets are the first assigned to a priority class. Each priority class has its own queue. The packets in the highest-priority queue are processed first. Packets in the lowest-priority queue are processed last. None that the system does not stop serving a queue until it is empty. Figure 002 shows priority queuing with two priority levels (for simplicity)

A priority queue can provide better Qos than the FIFO queue because higher-priority traffic, such a multimedia, can reach the destination with less delay. However, there is a potential drawback. If there is a continuous flow in a high-priority queue, the packers in the lower-priority queues will never have a chance to be processed. This is a condition called starvation.

3.1.3 WEIGHTED FAIR QUEUING

A better scheduling method is weighted fair queuing, In this technique, the packers are still assigned to different classes and admitted to different queues. The queues, however, are weighted based on priority of the queues; higher priority means a higher Wight. The system processes packets in each queue in a round-robin fashion with the number of packets selected from each queue based on the corresponding weight. For example, if the weights are 3,2, and 1, three packets are processed from the first queue, two from the second queue, and one from the third queue., If the system does not impose priority on the classes, all weights can be equal, In this way,

we have fair queuing with priority. Figure 003 shows the technique with three classes.

3.1.4 CLASS BASED QUEUING

CBQ is queuing algorithm that divides a networks connections bandwidth among multiple queens or classes.

A queue may optionally be configured to borrow bandwidth from its parent queue if the parent is being under utilized.

CBQ arranged in hierarchical manner. A top of hierarchical is the root queue which defines the total amount of bandwidth available. Child queues are created under the root queue, each of which can be assigned some portion of the root queues bandwidth.

3.2 TRAFFIC SHAPING

Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: leaky bucket and token bucket.

3.2.1 LEAKY BUCKET

If a bucket has a small hole at the bottom, the water leaks from the bucket at a constant rate as long as there is water in the bucket. The rate at which the water leaks does not depend on the rate at which the water is input to the bucket unless the bucket is empty. The input rate can vary, but the output rate remains constant. Similarly, in networking, a technique called **leaky bucket** can smooth out bursty traffic. Bursty chunks are stored in the bucket and sent at an average.

Figure.2 shows a leaky bucket and its effects.

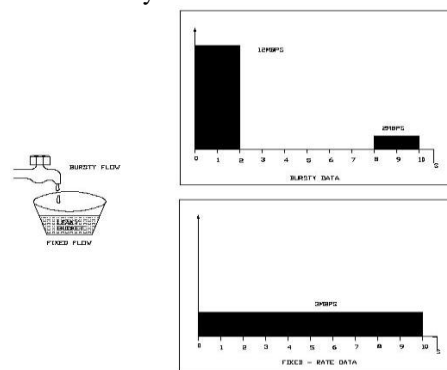


Fig-2. Leaky Bucket

In the figure, we assume that the network has committed a bandwidth of 3 Mbps for a host. The use of the leaky bucket shapes the input traffic to make it conform to this commitment. In Figure 005 the host sends a burst of data at a rate of 12 Mbps for 2 s, for total of 24 Mbits of data. The host is silent for 5 s and then sends data at a rate of 2 Mbps for 3 s, for a total of 6 Mbits of data. In all, the host has sent 30 Mbits of data in 10 s. The leaky bucket smooth's the traffic by sending out data at a rate of 3 Mbps during the same 10 s. Without the leaky bucket, the beginning burst may have hurt the network by consuming more bandwidth that is set aside for this host. We can also see that the leaky bucket may prevent congestion. As an analogy, consider the freeway during rush hour (bursty traffic). If, instead, commuters could stagger their working hours, congestion on our freeways could be avoided.

A simple leak bucket implementation is shown in Figure 005. A FIFO queue holds the packets. If the traffic consist of fixed-size packets (e.g., cells in ATM networks), the process removes a fixed number of packets from the queue at each tick of the clock. If the traffic consist of variable-length,

packets, the fixed output rate must be based on the number of bytes or bits.

The following is an algorithm for variable-length packets:

- ❖ Initialize a counter to n at the tick of the clock.
- ❖ If n is greater than the size of the packet, send the packet and decrement the counter by the packet size. Repeat this step until n is smaller than the packet size.
- ❖ Reset the counter and go to step 1.

3.2.2 TOKEN BUCKET

The leaky bucket is very restrictive. It does not credit an idle host. For example, if a host is not sending for a while, its bucket becomes empty. Now if the host has bursty data, the leaky bucket allows only an average rate. The time when the host was idle is not taken into account. On the other hand, the token bucket algorithm allows idle host to accumulate credit for the future in the form of tokens. For each tick of the clock, the system sends n tokens to the bucket. The system removes one token for every cell (or byte) of data sent. For example, if n is 100 and the host is idle for 100 ticks, the bucket collects 10,000 tokens.

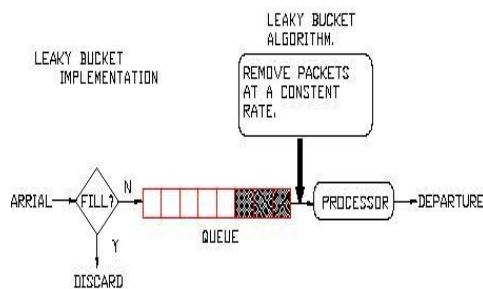


Fig-3. Leaky Bucket Implementation

Now the host can consume all these tokens in one tick with 10,000 cells, or the host takes 1000 ticks with 10 cells per tick. In other words, the host can send bursty data as long as the bucket is not empty. Figure.4 shows the idea.

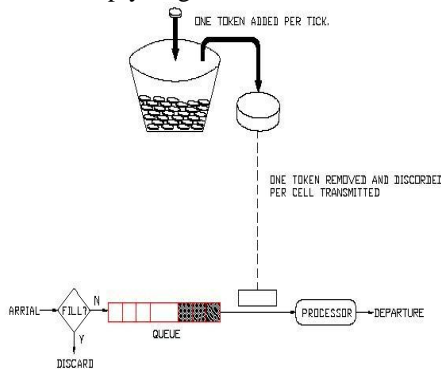


Fig.4 Token Bucket

The token bucket can easily be implemented with a counter. The token is initialized to zero. Each time a token is added, the counter is incremented by 1. Each time a unit of data is sent, the counter is decremented by 1. When the counter is zero, the host cannot send data.

3.2.3 WINDOW SIZE

Window size indicates the size of the devices receive buffer for the particular connection. In the window size represent how much data a device can handle from its peer at one time. For example serves window size was 360.

This means the serves is willing to take to home than 360 bytes at a time from a client.

3.2.4 MSS SHRINKING

TCP sender always chooses the largest possible packet size. Max segment size ,MSS typically ranging from 1000 to

1500 bytes such large packet require a long transmission time to be completely put on to the wire. When the WAN link bandwidth is narrow the long transmission time can block mission. Critical real time traffic such as VOIP, hence significantly degree the quality.

IV. CONCLUSION

As a whole Bandwidth Management is a powerful value added service. By using this value added service, we hope that the user traffic can be controlled & network resources could be used efficiently.

REFERENCES

- [1] N. Basher, A. Mahanti, A. Mahanti, C. Williamson, and M. Arlitt, "A comparative analysis of web and peer-to-peer traffic," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*. ACM, 2008, pp. 287–296.
- [2] H. Yun Wei, Y. dar Lin, N. Chiao, and T. University, "A survey and measurement-based comparison of bandwidth management techniques," *IEEE Communications Surveys and Tutorials*, vol. 5, p. 2003.
- [3] E. Bowen, C. Jeffries, L. Kencl, A. Kind, and R. Pletka, "Bandwidth allocation for nonresponsive flows with active queue management," *Broadband Communications, 2002. Access, Transmission, Networking, 2002 International Zurich Seminar on*, pp. 13–1–13–6, 2002.
- [4] G. Dias and C. Gunaratne, "Using dynamic delay pools for bandwidth management," *Proceedings of the 7th International Workshop on Web Content Caching and Distribution, Boulder, Colorado*, August 2002.
- [5] J. Valenzuela, A. Monleon, I. S. Esteban, M. Portoles, and O. Sallent, "A hierarchical token bucket algorithm to enhance qos in ieee 802.11: proposal, implementation and evaluation," *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 4, pp. 2659–2662 Vol. 4, Sept. 2004.



Mrs. N. Snehalatha is an Assistant professor in the Dept. of Software Engineering, SRM University, Chennai, Tamil Nadu, India. She has received her B.E in computer science engineering from Madras University, India & M.E in computer science engineering from Annamalai University, India. She has total 6 years of Teaching experience. She is persuing Ph.D in Wireless Network.



Dr. Paul Rodrigues is Dean Research, Vellammal University, Chennai, India and CTO of WisdomTree Software Solutions, Chennai, India. He has received his B.Tech from Karnataka University, India & M.Tech from NIT-Allahabad, India and Ph.D from Pondicherry University, India. He has total 20 years of Teaching and Industry experience in Delivery Management, Software Engineering, Budget Management and Business

Development. He has published more than 50 (refereed) papers in International Conferences/Journals which include Extreme Programming, Software Architecture, Databases and Object Oriented Analysis and Design. Also, he was an author of a chapter in the book "Recent Trends in Network Security & Applications" during July 2010 published by Springer Berlin Heidelberg publications, New York, USA. He is first in the world to apply Vastu to Software Architecture. He has worked in various domains that include Insurance, Retail, Digital Forensic, Content Management and Application Migrations. He is an active member of many professional-bodies like Identity Research Group, PMP, and CISSP.



Mrs. S. Angeline Julia is an Assistant professor in the Dept. of Software Engineering, SRM University, Chennai, Tamil Nadu, India. She has received her B.E in computer science engineering from Manonmaniam Sundaranar University, India & M.E in Embedded Systems Technologies from Anna University, India. She has total 6 years of Teaching experience. She is persuing Ph.D in Software Engineering.

Table: Comparison Of Bandwidth Management Techniques

METHOD	SCHEDULING					TRAFFIC SHAPPING				BUFFERING
TECHNIQUE/ PARAMETER	FIFO	PRIORITY	ROUND ROBIN	WEIGHT FAIR QUEUEING	CLASS BASED QUEUEING	WINDOW SIZING	MSS SHRINKING	LEACKY BUCKET	TOCKEN BUCKET	RED
Way of processing	First req. serves	High priority request process first	All the processors processed in their time slots	Processed based on priority & Round Robin	Processed based on Queue(class) sharing	Limiting the size of Message in TCP	Shirking MSS in TCP (Max segment size range from 1000 to 1500 bytes)	Shape burst traffic into fixed rate traffic	Each tick 'n' token to bucket remove one for every byte of data sent	RED uses a linear function of average queue length whether to drop/mark the incoming packet
Starvation	Occured	Occured	Not occured	Occured	Not occured	Occured	Occured	Occured	Occured	Low
Packet loss	Queue full the occured	Low priority process may lost	Very low	Very low	Low	Congestion occurred then loss happened	Low	Low	Very high	occured
Bandwidth guarantee	guarantee	Guarantee for high priority req.	guarantee	Problem for low priority req.	OK	---	---	OK	Average	guarantee
Delay	Occured	Low delay for high priority	Equal delay	Average	Average	High traffic delay happen	High traffic delay happen	Medium	Occured	Low delay
Scalability	Nil (for fixed queue)	Nil	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Accuracy	Good	Good	Low	Medium	Medium	Till good congestion	Good	Medium	Good	Good