ISSN: 2321-7782 (Online)
Volume 3, Issue 4, April 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

# Study of Analysis and Controlling Network Management System

**Monu Rani**[1]
Deptt. Of Comp. Sci. & Engg
SKITM, Bahadurgarh - India

**Dr. V. K. Pandey**[2]
Deptt. Of Comp. Sci. & Engg
SKITM, Bahadurgarh - India

***Abstract: Congestion in network occurs due to exceed in aggregate demand as compared to the accessible capacity of the resources. Network congestion will increase as network speed increases and new effective congestion control methods are needed, especially to handle "bursty" traffic of today's very high speed networks .In this research paper study of analysis and controlling techniques of the network congestion have been discussed. This paper discusses how we can control the network when it has been occurred.***

***Keywords: network analysis techniques and network controlling techniques, TCL, EEM (Embedded Event manager) scripting.***

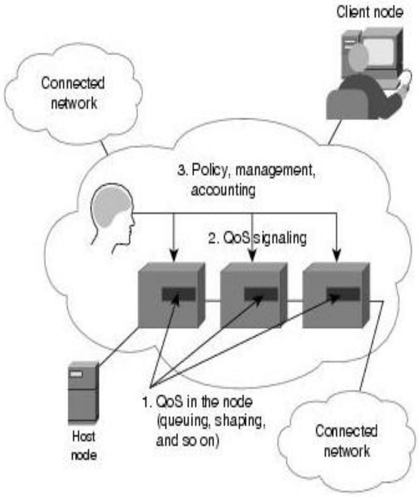
## I. Introduction

The Internet and wireless technologies are growing **I**rapidly and have been a tremendous success in the past few years. Its presence in everyday life is a fact. Traditional slow speed networks have been forced to merge with the high speed networks. But due to increase in Internet size and no. of users, clients are likely to experience longer delay, more packet loss and other performance degradation issues because of network congestion. Formally this problem was tackled by network service providers in terms of keeping Utilization of the network low, which may regard as an infeasible solution. As the Internet is gradually dominated by the IP and packet switching, so to increase the network performance in terms of satisfactory level of service to clients is considered as challenging problem. In today's Internet end systems, congestion control mechanism is performed at transports layer. Network traffic produced by these multimedia applications is known to be sensitive in nature and because of random queuing in routers there is a chance of occurrence of delay jitters and end to end delay. In most of the congestion control mechanisms, network routers are equipped with tail drop mechanism having finite capacity queue. When the server is busy, tail drop mechanism accommodate the incoming packets temporarily but upon queue full stage the arriving packets are dropped accordingly. Apart from simplicity, the technique may suffer various problems i.e. lockout behavior, global synchronization and full queue.

## II. Congestion Management

Congestion management techniques control congestion after it has occurred. One way that network elements handle an overflow of arriving traffic is to use a queuing algorithm to sort the traffic, then determine some servicing method of prioritizing it onto an output link.

Cisco IOS XR software implements the low-latency Queuing (LLQ) feature, which brings strict priority queuing (PQ) to the Modified Deficit Round Robin (MDRR) scheduling mechanism. LLQ with strict PQ allows delay-sensitive data such as voice, to be dequeued and sent before packets in other queues are dequeued. Cisco IOS XR software includes traffic policing capabilities available on a per-class basis as well as class-based shaping. The traffic policing feature limits the input or output transmission rate of a class of traffic based on user-defined criteria, and can mark packets by setting values such as IP Precedence, QoS group, or DSCP value. Traffic shaping allows control over the traffic that leaves an interface to match its flow to the speed of the remote target interface and ensure that the traffic conforms to the policies contracted for it. Thus, traffic adhering to a particular profile can be shaped to meet downstream requirements, thereby eliminating bottlenecks in topologi with data-rate mismatches.


Client node
Connected network
3. Policy, management, accounting
2. QoS signaling
1. QoS in the node (queuing, shaping, and so on)
Host node
Connected network

### *Access Node Control Protocol*

Access Node Control Protocol (ANCP) creates a control plane between a service-oriented aggregation device and an access node (AN) (for example, a DSLAM) in order to perform QoS-related, service-related, and subscriber-related operations. An ANCP Network Access Server (NAS) accepts and maintains ANCP adjacencies (sessions with an ANCP neighbor), and sending and receiving ANCP messages. ANCP allows static mapping between AN ports and VLAN sub interfaces so that DSL rate updates for a specific subscriber received by the ANCP server are applied to the QoS configuration corresponding to that subscriber . DSL train rates received via ANCP are used to alter shaping rates on subscriber-facing interfaces and sub interfaces on the router.

## III. Network Analysis And Sniffing

This paper proposes the IND-OCPA-P model to analyze the security of the proposed EOB and the encryption schemes supporting an efficient range query over encrypted data.

Network analysis(also known as traffic analysis, protocol analysis, sniffing, packet analysis, eaves dropping, and so on) is the process of capturing network traffic and inspecting it closely to determine what is happening on the network. A network analyzer decodes the data packets of common protocols and displays the network traffic in readable format. A sniffer is a program that monitors data traveling over a network. Unauthorized sniffers are dangerous to network security because they are difficult to detect and can be inserted almost anywhere, which makes them a favorite weapon of hackers.

A network analyzer is a combination of hardware and software. Although there are differences in each product, a network analyzer is composed of five basic parts:

» **Hardware** Most network analyzers are software-based and work with standard operating systems (OSes) and network interface cards (NICs). However, some hardware network analyzers offer additional benefits such as Analyzing hardware faults (e.g., cyclic redundancy check (CRC) errors, Voltage problems, cable problems, jitter, jabber, negotiation errors, and so on).

» **Capture Driver** This is the part of the network analyzer that is responsible for capturing raw network traffic from the cable. It filters out the Traffic that you want to keep and stores the captured data in a buffer. This is the core of a network analyzer—you cannot capture data without it.

» **Buffer** This component stores the captured data. Data can be stored in a buffer until it is full or in a rotation method (e.g., a round robin") where the newest data replaces the oldest data. Buffers can be disk-based or memory-based.

- **Real-time Analysis** This feature analyzes the data as it comes off the cable. Some network analyzers use it to find network performance issues, and network intrusion detection systems (ID Ses) use it to look for signs of intruder activity.
- **Decode** This component displays the contents (with descriptions) of the network traffic so that it is readable. Decodes are specific to each protocol, thus network analyzers vary in the number of decodes they currently support. However, new decodes are constantly being added to network analyzers.

## IV. Network Controlling Techniques

Congestion is a problem that occurs on shared networks when multiple users contend for access to the same resources (bandwidth, buffers, and queues). Think about freeway congestion. Many vehicles enter the freeway without regard for impending or existing congestion. As more vehicles enter the freeway, congestion gets worse. Eventually, the on-ramps may back up, preventing vehicles from getting on at all.

Congestion typically occurs where multiple links feed into a single link, such as where internal LANs are connected to WAN links. Congestion also occurs at routers in core networks where nodes are subjected to more traffic than they are designed to handle. TCP/IP networks such as the Internet are especially susceptible to congestion because of their basic connection- less nature. There are no virtual circuits with guaranteed bandwidth. Packets are injected by any host at any time, and those packets are variable in size, which make predicting traffic patterns and providing guaranteed service impossible. While connectionless networks have advantages, quality of service is not one of them.

## V. Major Performance Measures & Overview Of Congestion Control Schemes

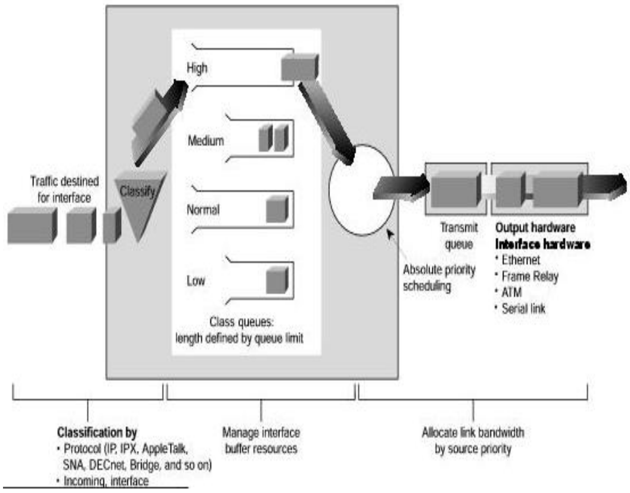
The major performance metrics under consideration are:

- Throughput
- Mean Queue length

The most widely deployed congestion control mechanisms are:

### *Drop Tail*

Drop tail is the simplest and most widely used congestion control scheme in the current Internet routers. It works on first-in-first out (FIFO) based queue of limited size, which simply drops any incoming packets when the queue becomes full. Because of its simple nature, it's easy to implement. Apart from simplicity other advantages include suitability to heterogeneity and its decentralized nature moreover its FIFO based queue provides better link utilization and it helps to absorb the bursty traffic.



### *Active Queue Management*

Active queue management is a technique in which routers actively drop packets from queues as a signal to senders that they should slow down. RFC 2309 lists the following advantages of active queue management

- Bursts are inevitable. Keeping queue size small and actively managing queues improves a router's ability to absorb bursts without dropping excessive packets.
- If a source overflows a shared queue, all the devices sharing that queue will slow down (the "global synchronization" problem).
- Recovering from many dropped packets is more difficult than recovering from a single dropped packet.

### *AIMD: Additive Increase/Multiplicative-Decrease*

In traditional TCP, the feedback control algorithm used to avoid congestion is the “additive increase/multiplicative-decrease (AIMD)”. This algorithm is basically used to implement TCP window. When congestion takes place, AIMD linearly expended congestion window with exponential decrease in it. The general rule of additive increase is to increase the congestion window by 1 maximum segment size (MSS) every round trip time (RTT) up to the detection of packet loss.

### *Random Early Detection (RED):*

RED algorithm for RED Gateways was first of all proposed by Sally Floyd and Van Jacobson [5], it calculates the average queue size by using a low pass filter with Exponential Weighted Moving Average (EWMA). RED addresses the shortcomings of traditional Drop Tail algorithm. Router using RED signals incipient congestion to TCP by dropping packets probabilistically before the queue becomes full and this drop probability is depending on running average queue size (qa).

### *Blue:*

Blue is another extension of RED developed by Wu-Chang and Feng et al] which uses packet loss and link utilization (rather than queue size) as a control variables to measure the network congestion.

### *ECN (Explicit Congestion Notification)*

The problem with RED is that it drops packets. A more efficient technique would be for a router to set a congestion notification bit in a packet, and then send the packet to the receiver. The receiver could then inform the sender to slow down via a message in the ACK. All the while, the receiver gets its packet and we avoid using packet drops to signal congestion.

ECN is an end-to-end congestion avoidance mechanism that adopts this technique. As the name implies, ECN provides direct notification of congestion rather than indirectly signaling congestion via dropped packets.

### *TCP Rate Control*

TCP rate control is a technique in which endpoints can adjust their transmissions based on feedback from network devices that perform rate control. Packeteer is an advocate of rate control and this section describes how the company implements it in its Packet Shaper products. Packeteer's Web site has numerous papers on rate control and other congestion control topics.

TCP Rate Control is also known as ERC (explicit rate control). A form of ERC is implemented in ATM networks. The Lawrence G. Roberts paper mentioned earlier in this section describes ERC in both ATM and TCP networks.

## VI. Embedded Event Manager (EEM)

***What is EEM?***

Cisco IOS Embedded Event Manager (EEM) is a powerful and flexible subsystem that provides real-time network event detection and onboard automation. It gives you the ability to adapt the behavior of your network devices to align with your business needs.

Your business can benefit from the capabilities of IOS Embedded Event Manager without upgrading to a new version of Cisco IOS Software. It is available on a wide range of Cisco platforms.

IOS Embedded Event Manager supports more than 20 event detectors that are highly integrated with different Cisco IOS Software components to trigger actions in response to network events. Your business logic can be injected into network operations using IOS Embedded Event Manager policies. These policies are programmed using either simple command-line interface (CLi) or using a scripting language called Tool Command Language (Tcl).

***EEM Types:***

There are two EEM independent pieces (types): Applets and Scripting

- Applets are a collection of CLI commands
- Scripts are actions coded up in TCL (interpreter language)

***EEM Actions can be:***

1. Sending an email messages
2. Executing a Cisco command.
3. Generating SNMP traps
4. Reloading the router
5. Generating priotized syslog messages
6. Switching to a secondary processor in a redundant platform
7. requesting system information when an event occurs(like sh tech,sh proccess cpu history)

***Applets versus Tcl Scripts versus Shell Policies***

In the years since the introduction of Cisco's Embedded Event Manager (EEM) many EEM policies have been developed inside and outside of Cisco. In the development of those policies many lessons have been learned about what works best and what does not. This document strives to outline some of the best practices that have been identified over the years when it comes to Cisco EEM policy design and development.

There are currently three native policy engines within Cisco EEM.

- Applets - Supported since EEM version 1.0, these policies are specified and defined in the configuration of the device and were designed to allow simple interface into Cisco's EEM feature.
- Tcl Scripts - Supported since EEM version 2.0, these policies are defined in separate files stored locally on the device and specified (registered) by adding a single configuration command. Tcl allows for more complex policies and some EEM features like timer subscribers are (currently) only supported in EEM Tcl.

- Shell Policies - Supported since EEM version 3.2, these policies are defined in separate files stored locally on the device and specified (registered) by adding a single configuration command. Shell policies utilize the IOS shell feature. Support for shell policies is currently (mid-2010) limited to some switching platforms.

## VII. EEM Script Using Tool Command Language(TCL)

Tcl is a scripting language with a simple and consistent structure. It is an interpreted language and it can be executed either by the shell program tclsh, which contains only the Tcl part, by the window shell wish which also contains the Tk toolkit, or from the Prolog top-level loop or a Prolog program. Tcl is a powerful scripting language that runs under Unix, Linux, VMS, DOS/Windows, OS/2, and MacOS (at least). It provides all the usual high-level programming features that we've come to expect from languages like the Unix shell, Awk, Perl, or Rexx,

It is commonly used for rapid prototyping, scripted applications, GUIs and testing. Tcl is used on embedded systems platforms, both in its full form and in several other small-footprint versions.

### *Features*

Tcl's features include:-

- Commands are commonly variadic
- Everything can be dynamically redefined and overridden.
- All data types can be manipulated as strings, including source code.
- Fully dynamic, class-based object system, TclOO, including advanced features such as meta-classes, filters, and mixins.
- Event-driven interface to sockets and files. Time-based and user-defined events are also possible.
- Variable visibility restricted to lexical (static) scope by default, but uplevel and upvar allowing procs to interact with the enclosing functions' scopes.

## VIII. Conclusion

The Internet and wireless technologies are growing rapidly and have been a tremendous success in the past few years. Its presence in everyday life is a fact. Traditional slow speed networks have been forced to merge with the high speed networks. But due to increase in Internet size and no. of users, clients are likely to experience longer delay, more packet loss and other performance degradation issues because of network congestion.

Congestion analysis and congestion controlling techniques are the way of management of network. By using these techniques network adminisitor can solve the problems occur due to congestion.

But these techniques take so much time and manpower consuming. For avoid this situation network administer use a scripting language, EEM scripting. EEM scripting provides us a simple and automatic way for controlling and analysis the congestion in net work. Harnessing the significant intelligence within Cisco devices, IOS Embedded Event Manager helps enable creative solutions, including automated troubleshooting, fault detection, and device configuration.

## References

1. V. Jacobson, “Congestion Avoidance and Control”, in Proc. ACM SIGCOMM, pp. 314–329, August 1988.
2. D-M. Chiu and R. Jain, “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks”, Computer Networks and ISDN Systems, vol. 17, pp. 1–14, 1989.
3. M. Allman and V. Paxson, “TCP Congestion Control”, Internet Engineering Task Force, RFC 2581, April 1999.
4. S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-Based Congestion Control for Unicast Applications”, http://www.aciri.org/tfrc/, June 2000.

5. S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, vol. 1, no. 4, Aug. 1993.

6. L. S. Brakmo, S. W. O’Malley, and L. L. Peterson, “TCP Vegas: New Techniques for Congestion Detection and Avoidance”, in Proc. ACM SIGCOMM ’94, August 1994.

7. M. Mathis and J. Mahdavi, “Forward Acknowledgement: Refining TCP Congestion Control”, in Proc. ACM SIGCOMM, August 1996. [23] C. Jin, D. X. Wei and S. H. Low “FAST TCP: Motivation, Architecture, Algorithms, Performance”, IEEE Infocom, Hong Kong, March 2004.

8. Ao Tang, Jiantao Wang, Steven H. Low, “Understanding CHOKe: Throughput and Spatial Characteristics”, IEEE/ACM Trans. Netw. 12(4): 694-707, 2004.

9. Ramakrishnan, K.K., and Jain, R., “A Binary Feedback Scheme for Congestion Avoidance in Computer Networks”, ACM Transactions on Computer Systems, V.8, N.2, pp. 152-181, 1990.

10. Dorgham Sisalem, Henning Schulzrinne, “Congestion Control in TCP: Performance of Binary Congestion Notification Enhanced TCP Compared to Reno and Tahoe TCP”, 1996. Proceedings of International Conference on Netwok k Protocols, pp.268 – 275, 1996.