

A Novel Architecture and Mechanism for Congestion Control in High Speed Network

Shilpi Kaura¹ and A. K. Vatsa²

^{1,2}School of Computer Engineering and Information Technology, Shobhit University, Meerut, (U.P), India

¹shilpi.kaura@gmail.com

²avimanyou@rediffmail.com

Abstract

Today is the era of multimedia applications such as audio or video transmission. Multimedia traffic can be classified according to bandwidth, latency and Quality of Service. For multimedia applications, there is strong need of higher transmission capacity and higher bandwidth, which leads the need of high speed network based architecture at the forefront of several current technology based networks. Due to the unpredictable fluctuations and burstiness of traffic flows within high speed network congestion can occur frequently and TCP will perform poorly in future high speed network. In this paper, we purposed the solution for the problem of frequent congestion occurrence in high speed network and it's model that has learning capability on the basis of round trip time, node's outlink capacity and average queue size and passes a signal to the sender to change its sending rate accordingly. Therefore, congestion occurrence in high speed network can be resolved.

Keywords

High speed network, Congestion, Adaptive Congestion Control, Congestion Window, RTT.

INTRODUCTION

1.1 High Speed Network:

For transmitting or receiving high bandwidth data such as video data we need a network with high network capacity and high bandwidth, such type of networks are known as high speed network. In general, any connection to the customer of 256kbit/s (0.25Mbit/s) or greater is more concisely considered as high speed network. High speed network is made up of cables such as fiber optics that support transmission of high speed data. Continuous media applications such as video and audio are sensitive not only to the packet loss probability but also to the correlation of packet losses. For multimedia applications, there is strong need of high speed network based architecture and mechanism.

1.2 Congestion:

Congestion in a network may occur if the load on the network- the number of packets sent to the network- is greater than the capacity of the network-the number of packets a network can handle.

Congestion happens in any system that involves waiting. Congestion in a network or internetwork occurs because routers and switches have queues- buffers that hold the packets before and after processing. It degrades quality of service and also can lead to delays, lost data. Congestion can be brought on by several factors. If all of a sudden, streams of packets begin arriving on three or four input lines and all need the same output line, a queue will build up. If there is insufficient memory to hold all of them, packet will be lost. This problem cannot be solved by increasing memory, because Nagle discovered that if routers have an infinite memory, congestion gets worse, not better. Slow processor can also cause congestion. If routers' CPUs are slow at performing the bookkeeping tasks required, queues can build up, even though there is excess line capacity. Similarly, low bandwidth lines can also cause congestion.

1.3 Congestion Control:

Congestion control refers to the mechanism and techniques to control the congestion and keep the load below the capacity. It is a mechanism that can either prevent congestion, before it happens, or remove congestion, after it has happened. The objective of congestion control is to maintain the number of packets within the network below the level at which performance falls off dramatically. Due to the unpredictable fluctuations and burstiness of traffic flows within high speed network congestion can occur frequently. So we need efficient congestion control technique. There are many mechanisms developed for congestion control:

1.3.1 Adaptive Congestion Control:

Adaptive congestion control is a mechanism with learning capability. This learning capability enables the mechanism to adapt to dynamically changing network conditions to maintain stability and good performance. In this a feedback is send to the sender to change sending rate, according to the current network conditions. It is scalable with respect to changing delays, bandwidth and number of users utilizing the network. ACP is characterized by its learning capability which enables the protocol to adapt to the highly dynamic network environment to maintain stability and good performance. This learning capability is materialized by a novel estimation algorithm, which 'learns' about the number of flows utilizing each link in the network. Previous experience in the design of congestion control algorithms, ([17-21]) has shown that at each link, the number of flows utilizing the link is necessary in order to maintain stability in the presence of delays. However, this parameter is unknown and time varying. It is thus necessary to develop algorithms which estimate this parameter using local information only [1].

1.3.2 Rate Control Protocol:

Rate Control Protocol (RCP) is a congestion control algorithm designed for fast download times. RCP is designed for the typical flows of typical users in the Internet today. RCP has two components: (1) End-host congestion control layer that sits between IP and TCP/UDP. During introduction, the end-host could adapt by testing for RCP at each end and along the path, falling back to TCP if need-be. (2) Each router maintains a single fair-share rate per link[14].

1.3.3 Explicit Congestion Control Protocol:

XCP is a window based congestion control protocol intended for best effort traffic. Senders maintain their congestion window and RTT and communicate this to routers via a congestion

header in every packet. Sender uses the feedback field in the congestion header to request its desired window increase. Routers monitor the input traffic rates to each of their output queues. Based on the difference between the link bandwidth and its input traffic, router tells the flows sharing that link to increase or decrease their congestion window[15].

1.3.4 Competitive and Considerate Congestion Control Protocol (4CP):

4CP - a new congestion controller that implements the farsighted strategy. The controller has the following features: (i) it chooses an additive-increase and inverse-decrease window adjustment in congestion avoidance; (ii) the inverse-decrease parameter is adapted very slowly targeting verification of a given loss-throughput formula over large timescale; (iii) the window is used as a bad phase detector and can assume negative values: it indicates bad (respectively, good) phase when it is negative (respectively, positive) and it is used as the congestion window in good phases; (iv) it is sender-only change of standard TCP, and thus it can communicate to any standard TCP receiver (easing the deployment); (v) no change to network routers or other network infrastructure is required[16].

1.4 Flow Control:

Flow control limits the amount of data transmitted by the sending transport entity to a level, or rate that the receiver can manage. At the transport level flow control will allow the transport protocol entity in a host to restrict the flow of data over a logical connection from the transport protocol entity in another host. TCP uses an end-to-end flow control protocol to avoid having the sender send data too fast for the TCP receiver to receive and process it reliably. Having a mechanism for flow control is essential in an environment where machines of diverse network speeds communicate.

1.5 Problem Identification:

For transmitting low speed data such as text and voice data, TCP works well, but there are strong indications that in high speed data transmission it will not perform efficiently. In networking literature many congestion control protocols have been proposed and their features and performance have been reported. Therefore, Multimedia data is highly bursty in nature. So, High speed network is used in multimedia application such as audio or video data transmission, And on the basis of literature survey we found that bursty data is the main reason for congestion in any network. It has also been shown analytically that as the bandwidth delay products increase, TCP becomes oscillatory and prone to instability[13]. Therefore in high speed network congestion is the main problem and we need an efficient mechanism for congestion control.

1.5 Paper Organization:

The paper is organized as follows. In chapter- I we have discussed introduction which covers introductory part of congestion, mechanism for congestion control and problem identification. On basis of literature review, chapter -II describes the related works. On basis of literature survey we proposed architecture and its mechanism in chapter III. Chapter IV consist of conclusion followed by chapter V that has future scope. Finally chapter VI has references used in our paper.

Related Work

In this chapter we will explain briefly problems and their solution that is related to congestion control in high speed network. We have studied so many papers. From all of them we only discuss some important issues related to our proposed model.

In [1], The problems occur in several protocols that were previously used for congestion control. TCP, a widely used protocol, work efficiently in low speed data network but in the case of high speed network, it gives poor performance. Previous protocols were fail to satisfy key design requirements of congestion control protocols, such as max-min fairness, high utilization, small queue sizes and no observable packet drops.

To fix that problem they design a new adaptive protocol that has an learning capability. In this a new packet header is defined that contains three field: Sender's RTT estimate, Desired sending rate and Congestion bit. When an acknowledgment is sent by the user, an appropriate values are set in these three fields. In realistic traffic scenarios comprising of a small number of long flows and a large number of short flows, ACP outperforms both TCP and XCP, even in the presence of random packet losses. ACP does not require maintenance of per flow states within the network and utilizes an explicit multi-bit feedback signaling scheme. To maintain stability it implements at each link a novel estimation algorithm which estimates the number of flows utilizing the link[1].

Paper [2], To improve network utilization and robustness for ATM networks there is increasing evidence that reactive congestion controls are necessary. However the large bandwidth-delay product and the difficulty in traffic modeling inherent in ATM networks makes design of effective reactive controls difficult. They discuss problems in preventive techniques of congestion control. They said that open loop (preventive approach) CAC schemes are sensitive to either over allocation of connections, or actual excess flow above the agreed parameters. Another weakness of open loop ("preventive") control is the assumed existence of an accurate model. Thus the effectiveness of these schemes relies on whether an accurate enough model of the system behavior can be found.

They propose to achieve this by integrating Connection Admission Control (CAC) and flow control and by making use of a novel control concept. They aim to use a combined feedback and feed forward adaptive predictive control system to maintain the QoS close to a reference QoS (target QoS value), irrespective of variations in traffic .They define two distinct groups of traffic: controllable (e.g. delay tolerant traffic, such as Available Bit Rate) and uncontrollable (e.g. delay intolerant traffic, such as variable bit rate video). This allows to introduce the concept of network controllability. Controllability is achieved by simply bounding the uncontrollable traffic, which becomes the sole role of CAC[1].

Paper [3], Vast majority of multimedia applications used currently in Internet based on UDP and RTP transport protocol. However, these protocols offers no means of quality of service control and cannot thereby guarantee any level of QoS. Due to the fluctuation in the network conditions, the inability of UDP or RCP to support QoS control renders multimedia applications.

In this paper, they present a new adaptation scheme based on RTP protocol. The scheme is in part similar to other schemes proposed for adapting application to the network state. In their scheme, however, they take the scalability issue into reconsideration. They present a new scheme called

the direct adjustment algorithm for adapting the transmission rate of applications to the congestion level of the network[3].

In [4], In recent years it has become clear that it can perform very poorly in networks with high Bandwidth Delay Product (BDP) paths. The problem stems from the fact that the standard TCP, AIMD congestion control algorithm increases the congestion window too slowly. Congestion control is an important component of a transport protocol in a packet switched shared network.

[4]This paper proposes Adaptive Layered Transmission Control Protocol (ALTCP), with a set of modifications to the congestion window response of TCP to make it more scalable in high-speed networks. ALTCP modifies the TCP flow by using the concept of virtual layers, such that the convergence properties and Round Trip Time fairness behavior is maintained similar to that of TCP.

[5]In the current Internet, the TCP transport protocol detects congestion only after a packet has been dropped at the gateway. However, it would clearly be undesirable to have large queues (possibly on the order of a delay-bandwidth product) that were full much of the time; this would significantly increase the average delay in the network. Therefore, with increasingly high-speed networks, it is increasingly important to have mechanisms that keep throughput high but average queue sizes low.

They said that the most effective detection of congestion can occur in the gateway itself. The gateway can reliably distinguish between propagation delay and persistent queueing delay. Only the gateway has a unified view of the queuing behavior over time; the perspective of individual connections is limited by the packet arrival patterns for those connections. In addition, a gateway is shared by many active connections with a wide range of roundtrip times, tolerances of delay, throughput requirements, etc.; decisions about the duration and magnitude of transient congestion to be allowed at the gateway are best made by the gateway itself [5].

Theory and experiments show that as the per-flow product of bandwidth and latency increases, TCP becomes inefficient and prone to instability, regardless of the queuing scheme. To address this problem, they develop a novel approach to Internet congestion control that outperforms TCP in conventional environments, and remains efficient, fair, scalable, and stable as the bandwidth-delay product increases. This new eXplicit Control Protocol, XCP, generalizes the Explicit Congestion Notification proposal (ECN). In addition, XCP introduces the new concept of decoupling utilization control from fairness control. Using a control theory framework, they model XCP and demonstrate it is stable and efficient regardless of the link capacity, the round trip delay, and the number of sources. Extensive packet-level simulations show that XCP outperforms TCP in both conventional and high bandwidth-delay environments. Further, XCP achieves fair bandwidth allocation, high utilization, small standing queue size, and near-zero packet drops, with both steady and highly varying traffic[15].

Paper[22] proposes a mechanism for equation-based congestion control for unicast traffic. Most best-effort traffic in the current Internet is well-served by the dominant transport protocol TCP. However, traffic such as best-effort unicast streaming multimedia could find use for a TCP-friendly congestion control mechanism that refrains from reducing the sending rate in half in response to a single packet drop. In their mechanism, the sender explicitly adjusts its sending rate

as a function of the measured rate of loss events, where a loss event consists of one or more packets dropped within a single round-trip time.

The basic decision in designing equation-based congestion control is to choose the underlying control equation. An application using congestion control that was significantly more aggressive than TCP could cause starvation for TCP traffic if both types of traffic were competing in a FIFO queue at a time of congestion [22].

[23] Their goal is to make realtime playback applications be good network citizen. A typical target application could be a web-server or a video-on-demand server that provides access to a variety of multimedia streams for a large number of heterogeneous clients. The idea is to separate congestion control from error control because the former depends on the state of the network while the latter is application specific. The server's transmission rate is continuously adjusted by the Rate Adaptation Protocol (RAP) in a TCP-friendly fashion. The RAP module is exclusively in charge of congestion control and loss detection. The layer manager adapts the quality of transmitted streams based on the rate specified by the RAP module. The layer manager tries to deliver the maximum number of layers that can fit in the available bandwidth. Rate adaptation happens on a timescale of round-trip times but layers are added and dropped on a longer timescale by using receiver buffering to accommodate temporary mismatches between transmission and consumption rates [23].

[24] In heterogeneous network, achieving congestion avoidance is difficult because the congestion avoidance is difficult because the congestion feedback from one subnetwork may have no meaning to sources on other subnetwork. He proposes using changes in round trip delay as an implicit feedback. Using a black box model of the network, they derive an expression for optimal window as a function of the gradient of the delay window curve.

The delay based approach proposed in this paper is, what they call, a black box approach. It treats the network as a black box, which does not give any explicit feedback. Black box schemes are zero network overhead schemes. The flow control, congestion control and congestion avoidance mechanisms, while essential for network, are actually overheads since they themselves consume very resource they are suppose to allocate. But in this approach the overhead is nearly equal to zero [24].

[25] In this paper, they demonstrate the existence of fair end-to-end window-based congestion control protocols for packets-switched networks with first come-first served routers. The protocols use only information that is available to end hosts and are designed to converge reasonably fast.

They propose a multiclass network model and a mapping between window size vector and the rate vector defined by the model. Since it is through the rates that fair sharing is defined and the protocol can control the window, the relation between the rate vector and the window size vector is important. Their algorithm uses the propagation delay, measured delay and window size [25].

PROPOSED WORK

Architecture of congestion control:

Our objective has been to design an effective window based congestion control model assuming availability of a feedback mechanism which allows an explicit exchange of information between the end users and the network.

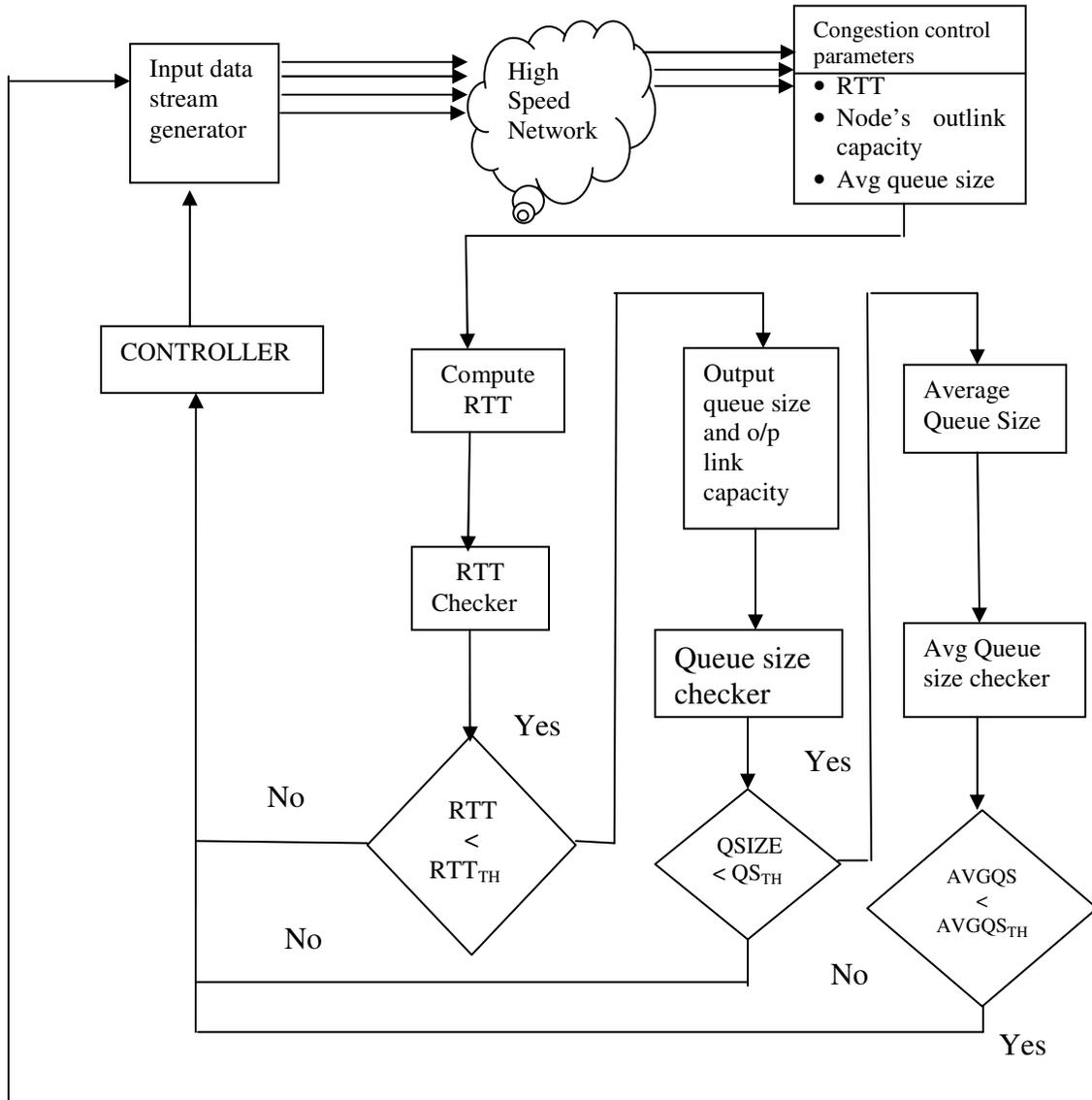


Figure 1: Architecture for congestion control

The purposed architecture has reactive approach in which when a feedback is passed to the sender of data, sender will update its sending rate accordingly, if required. The purposed architecture in Figure No. 1 components are defined as follows:

Input Stream Generator: Input stream generator is a source that produces multimedia data such as audio or video data which needs high bandwidth of the network. High speed network is needed to transmit such type of data.

Congestion Control Parameters: There are several parameters of any sending data which can help to control congestion in any network. We are considering four parameters for congestion control named as RTT(round trip time), output queue size, outlink capacity and average queue size.

RTT Checker: RTT checker is a module that will check whether the measured RTT is less then predefined threshold value of RTT or not and will take step accordingly.

Queue Size Checker: Queue size checker is same as RTT checker. It checks whether measured queue size is less then predefined threshold queue size or not and take action according to the result.

Average Queue Size Checker: This module compares calculated average queue size and stored threshold value of average queue size. When the value of average queue size succeeds threshold value then it takes action according to predefine rule.

Controller: This is brain of this model. When congestion occurs, any one of above three modules sends signal to controller showing that congestion may occur on the path. On the basis of that signal, controller will take corrective action to prevent congestion in the network.

Working of purposed Architecture:

This proposed architecture has adaptive learning capability. In this every time a feedback signal is passed to the source indicating the current network condition and signals the source to update its sending rate if required.

In this architecture the data generator generates multimedia data that requires high bandwidth, therefore high speed network is required for such type of transmission. So such data is fed into high speed network. There are many parameters by which we can identify and prevent congestion in any network. We are considering four parameters for this purpose: RTT, output queue size, outlink capacity and average queue size. There are predefined threshold value for RTT(RTTTH), Queue size(QSTH), Outlink capacity and Average queue size(AVGQSTH).

There are three different modules that work on these parameters separately: RTT Checker, Queue Size Checker and Average Queue Size Checker. Working of this architecture is as follows:

Step 1: Measure current round trip time (RTT).

Step 2: RTT checker checks whether RTT is less than RTT_{TH} or greater.

if ($RTT < RTT_{TH}$)

Go to step 3

else

sends a signal to controller to update sending rate of the source.

Step 3: Compute output queue size (QS).

Step 4: Queue size checker checks whether Queue size is less than QS_{TH} or greater.

if ($QS < QS_{TH}$)

Go to step 5

else

sends a signal to controller to update sending rate of the source.

Step 5: Measure current Average queue size(AVGQS)

Step 6: Average queue size checker checks whether AVQS is less than $AVGQS_{TH}$ or greater

if ($AVQS < AVGQS_{TH}$)

Sends a feedback to source that signals no congestion in the network.

else

sends a signal to controller to update sending rate of the source.

Controller receives signal from three different modules. So there are three different cases defining an algorithm for congestion control, on the basis of the sender of the signal.

CASE 1: When signal comes from RTT Checker

There may be two types of data: 1) Steady Data and 2) Transient data. So we are considering both cases for congestion control. There is a congestion window (CW) which represents number of outstanding packets in queue. Congestion window is updated every time a signal comes from RTT checker. It has initial value 1 and has W as maximum value. Whenever a new acknowledgment is received, a new desired sending rate is sent by the network which is later used for updating

congestion window. Initially an empty packet is send to check RTT. This process is done many times and then average of all RTTs are stored in the variable d. Later on this AVGRTT is updated after each acknowledgment. In the same way RTTMIN is calculated. The follows for this is as follows:

Initialization

**CW=1, CW_{max}=W;
MinSendingthreshold = STH_{MIN}, MaxSendingthreshold = STH_{MAX}**

for each arrival packet

```

if( STHMIN ≤ Sending Rate ≤ STHMAX )           // Steady State
{
    Congestion window is updated by
    desired sending rate × d
    CW=  $\frac{\text{desired sending rate} \times d}{\text{Size of packet(in Bytes)}}$ 
}

else if ( Sending Rate > STHMAX)                // Transient State
{
    Congestion window is updated by
    desired sending rate × RTTMIN
    CW=  $\frac{\text{desired sending rate} \times \text{RTT}_{\text{MIN}}}{\text{Size of packet(in Bytes)}}$ 
}
    
```

CASE 2: When signal comes from Queue size checker:

The objective of this algorithm is to match the input data rate (y) to the link capacity(C) and at the same time maintain small queue sizes. To achieve the latter the router maintains for each link, a value which represents the sending rate it desires from all users traversing the link. The desired sending rate is denoted by p and is updated every time a control timer expires. The control period is set equal to the average round trip time AvgRTT(d).There is a variable counter that increases by packet size each time a packet is received which is reset each time[1].

Mechanism is as follows:

Initialization

Control Period = d

Counter=0

For each packet arrival

Counter = Counter + packet size(in Bytes)

At the end of each Control Period

$$\text{Inputdata rate (y)} = \frac{\text{Counter}}{\text{Control Period}}$$

local queueing =instantaneous queue size / C

Propagation Delay = d- local queueing delay

q= minimum queue seen by the arriving packets during the last propagation delay.

The above variables are used to calculate the desired rate p every control period using the following iterative algorithm[1]:

$$p(k + 1) = Pr[p(k) + \frac{1}{\hat{N}(k)}[k_i(0.99 * C - y(k)) - \frac{1}{d(k)}k_q q(k)]], \quad p(0) = 0 \quad \dots\dots\dots$$

(1)

Where k_i and k_p are design parameters, \hat{N} represents an estimate of the number of flows utilizing the link and the projection operator is defined as follows[1]:

$$Pr[x] = \begin{cases} 0 & \text{if } x < 0 \\ C & \text{if } x > C \\ x & \text{otherwise} \end{cases}$$

.....(2)

The projection operator guarantees that the desired sending rate is non-negative and smaller than the link capacity [1].

CASE 3: When signals comes from Average queue checker:

In [5], The average queue size is compared to two thresholds, a minimum threshold and a maximum threshold. When the average queue size is less than the minimum threshold, no packets are marked. When the average queue size is greater than the maximum threshold, every arriving packet is marked. If marked packets are in fact dropped, or if all source nodes are cooperative, this ensures that the average queue size does not significantly exceed the maximum threshold. When the average queue size is between the minimum and the maximum threshold, each arriving packet is marked with probability p_a , where p_a is a function of the average queue size avg. Each time that a packet is marked, the probability that a packet is marked from a particular connection is roughly proportional to that connection's share of the bandwidth.

So mechanism is as follows:

Initialization:

$Avg \leftarrow 0$

$Count \leftarrow -1$

For each packet arrival

Calculate new avg queue size avg :

If the queue is nonempty

$avg \leftarrow (1 - w_q) avg + w_q q$

else

$m \leftarrow (1 - w_q)^{th} avg$

If $min_{th} \leq avg < max_{th}$

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

$count \leftarrow 0$

else if $max_{th} \leq avg$

mark the arriving packet

$count \leftarrow 0$

else $count \leftarrow -1$

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 pkt

Fixed parameters:

w_q : queue weight

min_{th} : minimum threshold for queue

max_{th} : maximum threshold for queue

max_p : maximum value of p

Others:

P_a : current pkt-marking probability

q : current queue size

$time$: current time

$f(t)$: a linear function of the time

In this way our proposed model checks all three parameters (RTT, node's outlink capacity, average queue size) and on the basis of current values of these parameters our proposed model finds whether there is congestion in the network or not.

CONCLUSION

In this paper, We have proposed Congestion Control architecture and mechanism for High Speed Network. Here, we have considered a High Speed Network in which Congestion usually occurs. Therefore, to avoid this congestion we adopt adaptive congestion control methods with few parameters values regulated through by instantiations condition of these parameters over High speed network. Parameters values may set or reset according to their value. The RTT, average queue size and channel's out link capacity regulates the node's transmission rate over Network that leads towards more effective and efficient mechanism for avoiding congestion over high speed network.

Future Scope

The proposed architecture and mechanism is effective and efficient but the more effective mechanism and architecture may be proposed as multimedia data burstiness can be decreased over high speed network in future that creates problems in high speed network in terms of RTT value, queue size value, node processing capability and existing channel bandwidth availability in high speed network.

REFERENCES

- [1] Marios Lestas Member IEEE, Petros Ioannou, Fellow IEEE Adaptive Congestion Protocol: A CongestionControl Protocol with Learning Capability
- [2] Andreas Pitsillides and Jim Lambert, Adaptive Congestion Control in ATM based Networks: Quality of Service and High Utilisation, JournalL of computer communications, 20, 1997, PP. 1239-1258.
- [3] Dorgam Sisalem, End to end Quality of service Control using adaptive Application, IFIP 1997, published by chapman & Hall
- [4] V.Kavidha, Dr. V.Sadasivam, Exploring Round Trip Time fairness for Adaptive Layered Transmission Control Protocol, Int. J. of Advanced Networking and Applications Volume:01, Issue: 06, Pages: 353-358 (2010)
- [5] Sally Floyd and Van Jacobson, Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, August 1993
- [6] ITU-T Recommendation I.371 (previously CCITT Recommendation), "Traffic Control and Congestion Contro in B-ISDN", March 1993
- [7] G. Gallassi, G. Rigolio, L. Fratta, "ATM: Bandwidth assignment and bandwidth enforcement policies", IEEE GLOBECOM'89, 49.6, pp. 1788-1793, 1989. 35
- [8] G. M. Woodruff, et al "A congestion control framework for high speed integrated packetized transport", GLOBECOM'88.
- [9] H. Suzuki, T. Murase, S. Sato, T. Takeuchi, "A simple and burst-variation independent measure of service quality for ATM traffic control", 7th ITC Specialist Seminar, 13.1, New Jersey, October 1990.
- [10] R. Guerin, H. Ahmadi, M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in highspeed networks", IEEE Journal of Selected Areas in Communications, Vol. 9, No. 7, September 1991.

- [11] E. Rathgeb, "Modelling and performance comparison of policing mechanisms for ATM networks", *IEEE Journal of Selected Areas in Communications*, April, 1991.
- [12] A. DeSimone, "Generating burstiness in networks: A simulation study of correlation effects in networks of queues", *Computer Communication Review*, Vol. 21, January 1991.
- [13] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J.C. Doyle. Dynamics of TCP/RED and a scalable control. In *Proc. IEEE INFOCOM*, volume 1, pages 23–27, June 2002.
- [14] N. Dokkipati, M. Kobayashi, Rui Zhang-Shen, and Nick McKeown. Processor sharing flows in the internet. In *Proc. Thirteenth International Workshop on Quality of Service 2005.*, June 2005
- [15] D. Katabi, M. Handley, and C. Rohrs. Internet congestion control for high-bandwidth-delay products. In *Proc. ACM SIGCOMM*, August 2002.
- [16] S. Liu, M. Vojnovic, and D. Gunawardena, "4CP: Competitive and aconsiderate congestion control protocol," 2006.
- [17] L. Benmohamed and S. M. Meerkov. Feedback control of congestion in packet switching networks: The case of a single congested node. *IEEE/ACM Transactions on Networking*, 1(6):693–708, December 1993.
- [18] C. E. Rohrs and R.A. Berry. A linear control approach to explicit rate feedback in ATM networks. In *Proc. IEEE INFOCOM'97*, volume 1, pages 277–282, March 1997.
- [19] T. Basar, E. Altman, and R. Srikant. A distributed globally convergent algorithm for fair, queue-length-based congestion control. In *Proc. IEEE Conf. Decision and Control*, volume 1, pages 622–627, December 2001.
- [20] Y. Zhao, S. Q. Li, and S. Sigarto. A linear dynamic model for design of stable explicit-rate ABR control systems. In *Proc. IEEE INFOCOM'97*, volume 1, pages 283–292, March 1997.
- [21] E. Altman, T. Basar, and R. Srikant. Robust rate control for ABR sources. In *Proc. IEEE INFOCOM'98*, volume 1, pages 166–173, March 1998.
- [22] Sally Floyd, Mark Handley, Jorg Widmer and Jitendra Padhye, Equation-Based Congestion Control for Unicast Applications, based upon work supported by AT&T, and by the National Science Foundation under grants NCR-9508274, ANI-9805185 and CDA-9502639, February 2000.
- [23] Reza Rejaie, Mark Handley and Deborah Estrin, RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet, supported by DARPA under contract No. DABT63-95-CoO95 and DABT63-96-C-0054 as part of SFT and VINT projects, IEEE 1999.
- [24] Raj Jain, A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks, Digital Equipment Corporation, April, 1989.
- [25] Jeonghoon Mo and Jean Walrand, Fair End-to-End Window – Based Congestion Control, *IEEE/ACM Transactions on Networking*, Vol 8, No. 5 October 2000.

Authors



Shilpi Kaura is working as Lecturer in the department of Computer Science and Engineering at Meerut Institute of Technology, Meerut (U.P.). She is pursuing M-Tech(Computer Engineering) from Shobhit University, Meerut. She obtained B-Tech(CSE) from Meerut Institute of Engineering and Technology, Meerut (U.P.). She has been in teaching for the past three years. She has been member of several academic and administrative bodies. She has attended several seminars and workshops.



A.K.Vatsa is working as Assistant Professor in the School of Computer Engineering and Information Technology at Shobhit University, Meerut (U.P.). He obtained his M-Tech(C.E.) from Shobhit University and B-Tech(I.T.) from V.B.S. Purvanchal University, Jaunpur (U.P.). He has worked as software engineer in software industry. He has been in teaching for the past one decade. He has been member of several academic and administrative bodies. During his teaching he has coordinated several Technical fests and National Conferences at Institute and University Level. He has attended several seminars, workshops and conferences at various levels. His several papers are published in various national and international conferences across India. His area of research includes MANET (Mobile Ad-Hoc network), Congestion Control, Network Security and VOIP-SIP (Voice over IP).