VOL. 10, NO. 9, MAY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ENHANCEMENT OF QoS IN MULTIMEDIA TRANSMISSION THROUGH OPTIMAL DELAY BASED FRAGMENTATION

S. Rajarajeswari[1] and J. Sutha[2]
[1]Department of Computer Science Engineering, Thamirabharani Engineering College, Tirunelveli, Tamilnadu, India
[2]Department of Computer Science Engineering, Sethu Institute of Technology, Virudhunagar District, Tamilnadu, India
E-Mail: srajeswari_me@yahoo.co.in

ABSTRACT

With the growth of different networking technology and multimedia technology the real time delivery of multimedia content becomes an imperative field. Most of the applications such as video conferencing need multimedia transmission techniques that send multimedia data from one end to another with enhanced efficiency in quality and minimized delay. Conventional packet fragmentation schemes shed a packet if all its fragments are not received correctly. But video data is loss tolerant and delay-sensitive. In this paper we propose a new family of delay based fragmentation algorithm which reduces the packet loss and delay thereby attain Quality of service in Multimedia applications.

Keywords: quality of service, delay, improved cuckoo search, fragmentation, improved chaotic bat algorithm.

1. INTRODUCTION

The requirement for real time data has augmented because of budding streaming technologies and availability of various portable devices that connect to internet for supporting multimedia content. The quality of service (QoS) requirement for delay, packet loss and throughput are considered as the Internet Protocol Performance Metrics. It refers to the capability of a network to provide better service to selected network traffic over various networking technologies. The primary goal of QoS is to provide priority including dedicated bandwidth, controlled Jitter, latency and loss characteristics. QoS for multimedia traffic takes into account several factors like balance between QoS parameters (Sufficient Bandwidth, end to end delay, throughput and packet loss), Deadline of multimedia packets, Packet loss, Nature of congestion, Size of packets and Reducing overhead caused by sorting of packets say Priority. Depending upon the handling of network traffic different applications have different requirements. For example real time data flows are time sensitive and delay intolerant whereas traditional data flows are loss intolerant but delay tolerant [1].

The most important characteristic of all real time traffic is that, the value of communication depends on the time of delivery of the message at the recipient. Video data is loss tolerant and a packet can be partially decoded even when some of its fragments are lost. But it is delay-sensitive (ie. Not delay tolerant) and retransmission of corrupted fragments may not be feasible. It is known that a deadline is associated with each message .A message is of no use if it arrives to the destination after its deadline has passed and those messages have to be discarded. A message that arrives earlier may be considered harmful as it requires buffering at the receiver to achieve constant end to end delay. Delivery at the recipient as close as possible to the deadline is sufficient and desired. Also more Packet loss in multimedia streaming degrades the quality of video and voice in the receiving end. Hence multimedia packets have to be properly fragmented, and scheduled to overcome the deadline violations, increased delay and packet loss.

2. RELATED WORK

Service differentiation is done based on CMSE, [2, 3] cumulative mean square error values to enhance the quality of pre encoded H.264/AVC compressed bit stream over bit rate limited error prone links. H.264 slices are prioritized based on CMSE (Cumulative mean square error) contribution towards video quality. Optimal fragment size is derived for attaining maximum good put. Low priority frames are dropped. However deadline is not considered and fairness of other data not considered. It is limited to video traffic efficiency.

Static priority with deadline considerations (SPD) [4]

This algorithm is the enhancement of Static Priority (SP) where the packets are given priorities based on the type of data they carry. If network node becomes congested these packets are sorted at the network node's queue according to their priorities. The packet with the highest priority of all queued packets is chosen next for transmission. The SP algorithm effectively provides a way to segregate applications into different classes.

The author proposes a packet scheduling algorithm which integrates QoS parameter, delay into classical static priority algorithm and analyze packet losses by considering buffer overflows and deadline violations. SPD works like SP but instead of complete sorting only k packets in the buffer are sorted. Service Differentiation is done based on the type of data that are carrying (ie. multimedia data have higher priority than email applications.

In SDP only k-sorting is performed where only first k packets are sorted thereby avoiding overhead. In SP the unfairness is caused because every packet arrival makes highest priority packet to move ahead leading to the starvation of low priority packets but in SPD only k-packets are sorted leading to the reduction of fairness. In SP deadline is not considered but in the case if SPD the packets with no remaining deadline is discarded thereby reducing the unnecessary traffic.

Timeliness and QoS aware packet scheduling [5]

Service differentiation is done based on QoS class. It may be either conversational or streaming application or non real time data .In this algorithm packets are assigned profit function based on the timeliness and network load status. It aims to maximize overall QoS metrics as different classes of data are treated based on the profit function. Fairness of other data, deadline and fragmentation not considered.

Competitive Scheduling of packets with Hard deadlines in a finite capacity queue [6]

The authors propose deterministic memory less algorithm and randomized memory less algorithm which aim to maximize the weighted throughput. Weighted throughput is the total value of transmitted packets by their deadlines. Service differentiation is done based on hard deadlines.

Dynamic priority packet scheduling (DPD) [7]

This algorithm integrates the delay parameter into the classical static priority algorithm .It selects two threshold levels that fixes the priority. When threshold increases this algorithm behaves as FIFO and When decreases it behaves as static priority algorithm SPD. The important characteristic of this algorithm is it explicitly takes the parameter deadline combined with priority. It modifies the packet priority depending upon the waiting time in the corresponding queue. It reduces the complexity and starvation, providing fairness to applications with different priorities.

Size oriented dropping policy (SDP) [8]

In SDP Service differentiation is done based on size of packets. This scheme differentiates real time and non real time applications using packet size. Transmission delay of a packet is directly proportional to the packet size. Hence small packets are preferably used by applications that require fast delivery times and big packets are preferred for bulk data transfers. SDP increases the apparent quality in real time applications. Size oriented dropping policy has been optimized by incorporating the dropping of packets based on optimal packet size [9]. In this system the authors mainly focuses on throughput and delay is not considered.

Applications with different characteristics have different requirements and their packets should be treated differently. For example real time data flows are time sensitive and delay intolerant whereas traditional data flows are loss intolerant but delay tolerant.

Service Differentiation can be done by various schemes. Based on the type of data that are carrying (i.e) video application has higher priority than an email application., Based on QoS class (conversational, Streaming, non real Time),Based on the CMSE values.(cumulative mean square error which takes into consideration the error propagation),Based on deadline, Based on size of packets .

Non Congestive Queuing NCQ+[10,11], improved version of NCQ[12] uses a second level of prioritization based on their impact on total delay. It can be adjusted to provide guaranteed services in terms of delay and improves the performance of applications as long as there is no violation of guaranteed services and other flows impact on the performance is insignificant.NCQ+ satisfies more users with diverse demands on delay and throughput.

Pseudocode Non-Congestive Queuing+

```
for every received packet
   begin
   count received packets
   if(packetlength<=size_thresh1) and
   (total_prioritised _service<ncqthresh1)
   Then
   Packet receives high priority
   Count favoured tiny packets
   Else
Ncqthresh2=ncqthresh1-k*tiny_packets_favor_rate
   Packet receives normal priority
   End.
If (size_thresh1< packet_length <=size_thresh2) and
(small packets_favor_rate<ncqthresh2) and
   (total_ prioritised _service<ncqthresh1)
   Then
   Packet receives high priority
   Count favoured small packets
   Else
   Ncqthresh2=ncqthresh -
    k*tiny_packets_favor_rate
   Packet receives normal priority
   End.
   End.
```

3. PROPOSED SYSTEM

The proposed system differentiates service based on size and focuses on the fragmentation of multimedia packets into optimal size using optimization techniques by considering both throughput and delay parameters. The packet size for which maximum amount of input reaches the destination with delay diminution is considered as optimal packet size. The scheduling and discarding policy is done based on this computed optimal size.

a) Packet fragmentation

Packet size is the most typical and easy to extract indicator of the type of application that created the packet [13].In real life, differentiation of packets is not binary; we cannot consider only small and big packets. Both 100 and 800 byte packets are smaller than a 1000 byte packet, yet we should not favor equally these two packets over the bigger one. Classification among ‘big’ and ‘small’ packets should be relative to the size of packets that already have been served by the router. For example a 500 byte packet

ARPN Journal of Engineering and Applied Sciences

is considered small from a router that recently served 1KB packets and big from a router that recently served 100 byte packets. The classification of packets as big and small by setting a single or static threshold value would also result in undesirable behavior. Also fragmentation of packet calls a trade off between reducing the number of overhead bits per packet by using large fragment size and reducing transmission error rate by using small packets. Hence optimal packet size for which maximum amount of input data reaches the destination in minimal time has to be computed to balance the above factors.

The formula for obtaining throughput is given below.

$$T = \frac{1}{\sqrt{2\pi}} e^{(f-nf)^2}/2 \left[\frac{\frac{(\min(f,(x-nf)))(s-h)}{s} + nf}{tot\_data} \right]$$

Where
x = channel rate /bandwidth
n = percentage of other data
s = optimal packet size
h = header size of packet
tot_data = total data (multimedia data + other data)
f = encoding rate of multimedia data.

The objective is to find optimal packet size 's' which maximizes the throughput T .Optimal packet size for different encoding rates are calculated using optimization techniques which considers the delay parameter for its quick convergence. The delay parameter is considered by integrating NCQ+ algorithm which do not contribute to congestion because of their appropriate packet sizes and low transmission rate. Hence the packets are fragmented with the scheme that the optimal packet size calculated using optimization technique not only increases the throughput but also decreases the delay .

b) Packet dropping

This is done similar to [9] using SDP and Modified RED where the packets are dropped based on optimal size and deadline.

c) Calculation using optimization technique

Techniques that reduces both computation time and materials /resource consumption to achieve optimum solution for a given problem. Optimization techniques are capable of finding optimum solution to problems for which exact and analytical methods do not produce optimal solution within a reasonable computation time. Several optimization techniques using mathematical and metaheuristic algorithms were proposed. The latter emerged as effective tool for attaining optimization . The proposed system computes optimal packet size using two techniques Improved Cuckoo Search (ICS) and Improved Chaotic Bat algorithm (IBACH) and selects the optimal value obtained in minimal time. Both the techniques takes the value as calculated based on delay parameter as its initial value for quick convergence.

d) Improved Cuckoo search [14]

Each egg in a nest represents a solution, and a cuckoo egg represents a new solution.

The aim is to employ the new and potentially better solutions (cuckoos) to replace not-so-good solutions in the nests. In the simplest form, each nest has one egg CS is based on three idealized rules: Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest; The best nests with high quality of eggs (solutions) will carry over to the next generations. The number of available host nests is fixed, and a host can discover an alien egg with probability P ε {0,1} .In this case, the host bird can either throw the egg away or abandon the nest to build a completely new nest in a new location.

Pseudocode

Begin
Objective function f(x),x=(x1....xd) $^{T.}$
Generate initial population of n host nests
Xi(i=1,2,...n)
While(t< Maxgeneration) or (stop criterion)
Get a cuckoo randomly by Levy flights
Evaluate its quality / fitness Fi.
Choose a nest among n randomly
If (Fi > Fj)
Replace j by the new solution.
End if.
A fraction (Pa) of worse nests are abandoned and new ones are built.
Keep the best solutions. (or nests with quality solutions)
Rank the solutions and find the current best.
End while
Post process results and visualization.
End.

e) Improved Chaotic Bat algorithm [15]

This is an improved version of a Bat Meta-heuristic Algorithm, that obtains the optimal solution within less computation, which save time in comparison with the traditional methods.

Step-1: Set the initial conditions: population xi (i = 1, 2 ...n) and Vi, pulse frequency fiat xi and pulse rates ri and the loudness Ai.
Step-2: Calculate the average position and the optimal position of the bat colony.
Step-3: Update velocities and locations/solutions and Generate new solutions by adjusting frequency.
Step-4: If (rand > ri) then select a solution among the best solutions and generate a local solution around the selected best solution .If not, skip this step.
Step-5: If (rand < Ai & f(xi) < f(x))then accept the new solutions. Increase ri and reduce Ai.If not, skip this step.
Step-6: Rank the bats and find the current best X.
Step-7: If the iterations attain to the maximum number, then stopped and output the global optimal solution. If not, go to step 2 to continue the search.

VOL. 10, NO. 9, MAY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

4. OUTCOMES

The optimal packet size is more or less the same for the above discussed two optimization techniques. Hence the average packet size is considered as optimal packet size and is listed in Table-1. The throughput for different packet sizes is depicted in Figure-1. The graph shows that maximum throughput is achieved when the packet size is optimal. Similarly Figure-2 depicts that minimum delay is achieved with optimal packet size.

Table-1. Optimal packet size computed using ics and ibach algorithm.

S. No.	**Encoding rate** (in Kbps)	**Optimal packet size** (in bytes)
1	384	150
2	512	168
3	640	184
4	768	196
5	896	208
6	1024	220
7	1152	228
8	1330	240
9	1408	244
10	1536	252

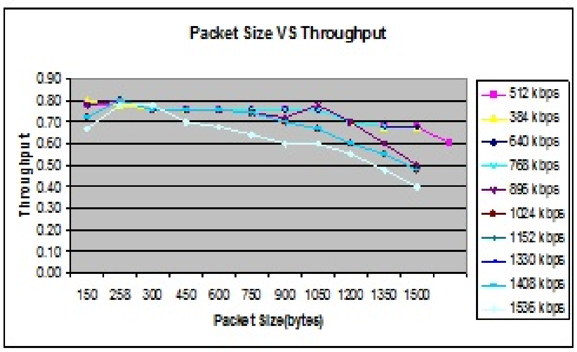

Figure-1. Packet size vs throughput for different encoding rates.

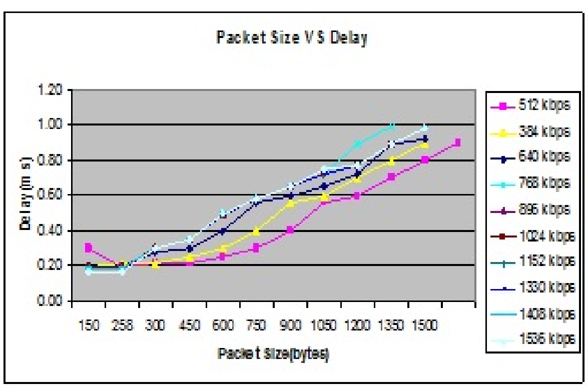

Figure-2. Packet size vs delay for different encoding rates.

5. CONCLUSIONS

The system concentrates on integrating well suited service differentiation scheme with proper dropping policy in minimal time to minimize the individual packet delay and the overall transmission delay and also improves throughput.The optimal fragment size is calculated for different encoding rates using Improved Cuckoo search and Improved Chaotic Bat algorithm. Experimental results reveal that this system suits more appropriately for delay sensitive applications rather than packet loss intolerant applications. Future work is to integrate appropriate scheduling mechanism and priority consideration to minimise packet drops.

REFERENCES

[1] José Neuman de Souza and Raouf Boutaba. 2000. Managing QoS in Multimedia Networks and Services Springer.

[2] P. C. Kambhatla, K. K. R. Kumar, S. Paluri, S. Cosman. 2012. Wireless H.264 Video quality enhancement through optimal prioritized packet fragmentation.

[3] http://code.ucsd.edu/cosman/PaluriICIP2012.pdf,2012.

[4] Tamer Dag. 2007. Static priority with deadline considerations: packet scheduling algorithm for achieving better QoS, ICNS.

[5] Yao-Nan Lien, Yung-Chuan Wun. 2009. Timeliness and QoS aware packet scheduling for next generation networks, International Conference on Network-Based Information Systems.

[6] Fei Li. 2009. Competitive Scheduling of Packets with Hard Deadlines in a Finite Capacity Queue, INFOCOM 2009, IEEE Digital Object Identifier.

[7] T. Dag. 2010. Dynamic Priority Packet Scheduler with Deadline Considerations (DPD), WMSCI’10, Orlando, FL, July.

[8] Stylianos Dimitriou, Vassilis Tsaoussidis and vassils Tsaussidis. 2010. Introducing Size oriented dropping policies as QoS supportive functions IEEE.

[9] K. Sasikalarani and J. Suganthi. 2014. Improving Quality of Service in IP Networks for Multimedia Applications with Optimal Fragmentation. Journal of Computer Science.

[10] G. Papastergiou, C. Georgiou, L. Mamatas and V. Tsaoussidis. 2008. On shortest Path first: A delay – oriented Prioritization policy.

[11] Lefteris Mamatas. 2010. Vassilis Tsaoussidis Less Impact Better Service (LIBS) A service paradigm for Internet telephony, Springer.

[12] Lefteris Mamatas and Vassilis Tsauossidis. 2008. Differentiating services with Non – Congestive

Queuing (NCQ), IEEE.

[13] S. Dimitriou, V. Tsaoussidis. 2008. "A new service differentiation scheme:size based treatment "June.

[14] Ehsan Valian, Shahram Mohanna and Saeed Tavakoli. 2011. "Improved Cuckoo Search Algorithm for Global Optimization" , Int. Journal of Commu. and Info. Tech.

[15] Osama Abdel-Raouf. 2014. Mohamed Abdel-Baset and Ibrahim El-Henawy, An Improved Chaotic Bat Algorithm for Solving Integer Programming Problems, International Journal of Computational Engineering & Management, March.