International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.2 (2014), pp.15-26
http://dx.doi.org/10.14257/ijmue.2014.9.2.02

Research of Proxy Cache Algorithm in Multi-media Education System

Zhang Yuhong[1], Gu Liwei[1] and Wang Yubin[2]

[1]College of Education, Hebei Normal University of Science & Technology, Qinhuangdao, Hebei, P.R.China , 066004
[2]College of Mathmatics and Information Science & Technology, Hebei Normal University of Science & Technology, Qinhuangdao, Hebei, P.R.China,066004

13933690306@163.com, xiaxue826@sohu.com, qhdbjy@126.com

Abstract

Multi-media education system is more and more widely used in all levels of education. In order to decrease cost of multi-media system and keep efficiency with increasing multi-media materials, proxy cache algorithm has been widely studied. Based on analysis of existing research of proxy cache results, an improved proxy coaching strategy of prefix cache and postfix merging is proposed. The strategy can dynamically adjust prefix cache size with the object access change. A more effective method of steaming merging has been proposed with multicast used in postfix portion. The results show that the improved strategy can effectively utilize proxy cache resource, shorten time delay and save band width.

Keywords: *multi-media education system; proxy cache; prefix cache; stream merging*

1. Introduction

Multi-media makes great change to human's life. It is a comprehensive electronic and information technology and greatly changes the trend of computer system, video and audio equipments. Multi-media technology can also be used in education. The technology can help students to concentrate on learning and improve efficiency. Visualization technology can use vivid 3D picture or video instead of abstract and boring lessons to improve intuition.

Multi-media has been used for education for years. There are also lots of researches on it. In the beginning, there are both internal and external problems on staff and students to resolve [1]. As the research development, Multi-media education system is divided into four types: linguistic visual, nonlinguistic visual, linguistic audible, and nonlinguistic audible [2]. Recent years, the develop direction has been studied [3].

With the development of distributed system, education system is integrated into school network and terminal computers as a conception has been used in education. In order to cut cost, all resources can be stored in the server and each computer in classroom or other places in school can be seen as terminal computer. In the system, data transmission can use streaming media and each computer accesses to the server to request data package. Data transmission has characteristics of high band width, short time delay and low jitter. So, it demands high quality of network and better data transmission system. With the increasing of the data and terminal computers, the access would be more frequently at a short time, for example, at the beginning of class. In order to make the efficiency higher and keep multimedia file play smoothly, the access algorithm should be studied to make full use of the bandwidth and multi-media resources.

ISSN: 1975-0080 IJMUE

Recently, many algorithms based on proxy has been proposed, such as prefix merging [4], uniform segmentation [5], exponential segmentation [6], layered media cache [7], multiple version cache [8], batching [9], patching [10], stream merging [11-14], and so on. In teaching system, the prefix part can be pre-stored in proxy server before lesson starting. This can reduce the start delay in the terminal computer to keep the teaching going well. In addition of patching and steaming merging technology [15-17], more terminal computers would share a same multicast streaming to effectively reduce the band width and server resource consumption but to get better performance. When one video or file has a high request frequency, algorithms described before would still consume high bandwidth and has little dynamic adjusting capability.

The algorithms described before just can be used in off-line state, but how to effectively use in proxy server cache condition is still to be resolved. Therefore this study intends to develop a new request algorithm between terminal computers and sever. This algorithm would be called CRMT (compatible reachable merge target) algorithm. Existing technology analysis is presented in Section 2. The CRMT algorithm is presented and discussed in Section 3. Simulation and results are showed in section 4 and concluding remarks are then given in Section 5.

2. Existing Technology Analysis

2.1. System description [18-20]

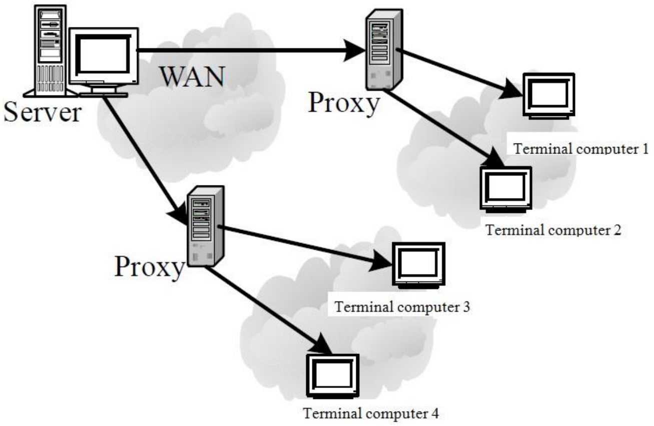

Figure 1. Proxy based streaming multimedia system

In order to describe the algorithm intuitively, we firstly give the system model as showing in Figure 1. All the terminal computers can be seen as a client cloud or divided into several client clouds. For example, four terminal computers can be seen as one client cloud or two client clouds. If it has been divided into two client clouds, terminal computer 1 and terminal computer 2 is included in client 1 while terminal computer 3 and terminal computer 4 is in client 2. There is a proxy between server and clients and all clients would request video or other documents from the server through the proxy. Parts of video or other files are stored in the proxy. We firstly make some assumption as the following:

(1) All the video request is from the begin of a special video;

(2) All the video transfer rate is same as that of playback;

(3) Each terminal computer just can receive five video streaming at one time.

In Figure 1, some parameters are defined. N is the number of multimedia files in server; $L(i)$ is the time length of file i; $Lc(i)$ is the time length of file i storage in proxy. $r(i)$ is the code rate of file i; $\lambda(i)$ is the request rate of file i; S is the proxy cache memory capacity; $Bc(k)$ is cache capacity of terminal computer K; $g(i, m)$ is the cache gain of part m of video i; $d\ \max$ is the maximum time delay between server and proxy.

2.2. Proxy based multimedia data transmission [21-23]

The video stream is divided into two parts: data transmission between server and proxy; data transmission between proxy and terminal computers. Due to the simple system, unicast is used in the data transmission between server and proxy. In the system, each proxy just communicate with server and terminal computers. Mixed data transmission ways of unicast and multicast are adopted in data exchange between proxy and terminal computers.

Assuming that $Bc = \min\{Bc(k)\}$, and Bc is the maximum cache memory capacity. There are three conditions in data transmission strategy. Detailed description is showed as the following:

(1) If $Bc \le Lc(i) \le L(i)$, as showed in Figure 2.

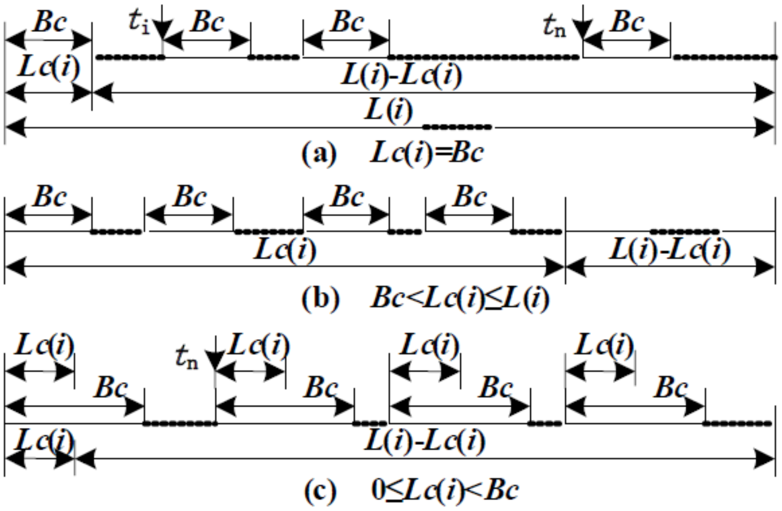

Figure 2. Some conditions

Define a request arrival time as the start time and Bc as the time length to be zone of Bc. Starting time of current zone of Bc is defined as Tb. When a new access arrives at time t, if $t - Tb \le Bc$, then, the access would belong to zone of Bc; or else, the access is not belong to the current zone and go into next zone at the start time t. With the same method, accesses arriving at different time belong to different Bc zones. To the whole video, whole time axle would be divided into every Bc zone. At the beginning of arbitrary zone of Bc, the proxy prepares a multicast streaming with time length of $Lc(i)$. In the Bc zone, it starts to receive video stream as the first access arrives when other requests arrives. As for each request in the Bc zone, with the t meets the condition of $t = Tb + Lc(i)$, all accesses would receive the multicast video part $(Lc(i), L(i)]$. The video part firstly transmits from server to proxy, and then, from proxy to the terminal computers in multicast ways.

It is obviously that bandwidth of video i can be showed as the following:

$$B_1(i) \leq \frac{\lambda(i) \times r(i)}{1 + \lambda(i) \times Bc}[L(i) - Lc(i)] \tag{1}$$

(2) If $0 \leq Lc(i) \leq Bc$, as shows in (c) of figure 2.

This can be divided into two conditions: 1) when access time t meets $t \leq Lc(i)$, the transmission strategy would be the same as $Bc \leq Lc(i) \leq L(i)$; 2) if access arriving time t meet the condition of $Lc(i) < t \leq Bc$, the access would receive and cache the part of video $[t, L(i)]$, at the same time, unicast is used to get the video $[0, Lc(i)]$, after $t + Lc(i)$, the video part $(Lc(i), t)$ would be got from server by unicast ways.

It is obviously that the video transmission would take bandwidth:

$$B_2(i) \leq \frac{\lambda(i) \times r(i)}{1 + \lambda(i) \times Bc}[L(i) - Lc(i)] + \frac{\lambda(i)\,(Bc - Lc(i))^2}{2} \tag{2}$$

2.3. Existing technology analysis

Local cache strategy divides the streaming multimedia file into lots of segments. The proxy server pre-store some of them. In local cache strategy, segment dividing has two typical methods. One is time zone based method and the other is mass based method. Time zone method can also be seen the same as segment based proxy cache strategy. There are also three typical methods: prefix cache, mean segments dividing and exponential segments dividing. All the method is the start part of the cache streaming multimedia. Typical mass zone based local cache strategy includes layered cache and multi version cache. Layered cache needs the streaming multimedia in layered coding. Multi version cache store various kinds of version and transmit special version with respond bandwidth.

Typical cache strategy is good for improving proxy performance, but mass based layer cache needs online code process. If a file is requested in a short time, CPU resource would be excessively consumed. Multi version is a simple cache method, but due to its cache of different versions, cache memory is wasted. In addition, the versions is limited and it can't adapt every condition in the network. Time based cache strategy has little concentration to take popularity and access mode of multimedia into consideration. Segments size and dividing method lack adaptability. Size of segment dividing too big would waste cache memory and too little would increase complexity of space dispatching.

The transmit strategy proposed can be divided into reaction type and non-reaction type. In reaction type, terminal computers send request to server or proxy. After they get the access, they would send message to terminal computers. In non-reaction type, server or proxy send data or information to terminal computers without any requests.

Prefix cache is pre-store the prefix of multimedia object in proxy. The proxy can be fetched from proxy directly to decrease the time delay in terminal computers. The proxy fetch the postfix from server while it deliver the prefix to terminal computers. This method can save bandwidth for the reason of using prefix deliver channel. In the method, it is quite important to determine size of prefix and segment dividing method. The two factors can greatly influence performance of system. It has many dividing method. In each dividing method, uniform dividing method divide the whole file into segments with same size, exponential dividing method dived the whole file into segments, which latter segment is twice size of the former one.

Key to merge streams is to establish an effective merging tree and the merging tree can merge terminal computer's request into multicast stream. The aim is to ensure the merging target and merging sequence. All the effort is to decrease the total bandwidth demanded by the server. The best method is optimal stream merging and early merging. The optimal method is according to calculation to find the merging tree with minimum resource consumption. Example is showed in Figure 4. Early merging algorithm is according to find nearest merging point to merge the targets. Take Figure 4 as an example, if in early merging as the algorithm, the results would change. The detail results is showed in Figure 3.

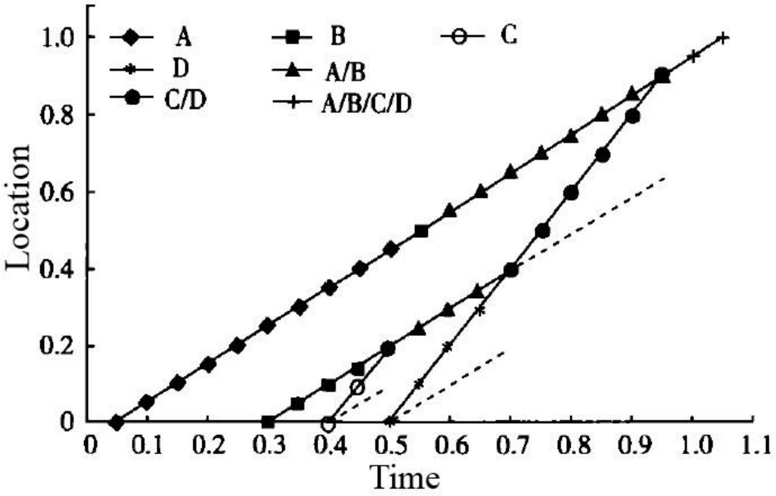

Figure 3. Early merging

3. CRMT

CRMT is derived from skyscraper algorithm and piggybacking algorithm. It has the same assumption with the system model. The network just has unicast between the server and proxy, and has both unicast and multicast between proxy and terminal computers. Terminal computers always request from start part of the video and proxy storages prefix to diminish the time delay of request of each terminal computer.

The algorithm is divided into two steps:

(1) Prefix of streaming multimedia is stored in proxy. This can provide the real time service. Capacity of segments and cache memory can be determined with the method later. It can decrease the effect of streaming multimedia on proxy efficiency.

(2) When all the requests have arrived at one cache zone, proxy would transmit the left data through the pre assigned postfix channels. This algorithm can help to decrease bandwidth of the network. The merging method in postfix data transmission can further reduce the network resource consumption.

3.1. Determination of segment time length

The segment time length should be determined step by step and adjusted dynamically. Here, we assume a segment time length s_i to be 20s. s_i is determined by number of streaming video and capacity of proxy server cache. Use N and S to represents number of streaming video and capacity of proxy server cache. When streaming multimedia is completely played, average request interval of terminal computers is used as an index to determine the time length of segment. Here, we use the mean time as the initial value of time length of segment. This initial time can be adjusted with the request arrives to meet self-demand. Some important parameters are showed in Table 1.

Table 1. Important parameters

Parameters	Definition	Unit
N	Number of streaming video	
S	Capacity of proxy cache	B
λ_i	Frequency of video i being request	Hz
r_i	Mean time length of video i	s
s_i	Segment length of video i	s

In the operation, if request frequency λ_i to video i increases, capacity of cache would adaptively increase. On the contrary, if λ_i decreases, capacity of responsible cache would decrease.

3.2. CRMT

In order to decrease the complexity and variation of merging tree, the paper proposes an algorithm of compatible reachable merge target (CRMT) to merge the postfix multicast streaming in the proxy cache system.

When a new request receives, the system would traverse the multicast with time sequence to find the nearest target streaming. If the traversal finds more than one merging points, all the points are compared to select the original one. Merging points, with shorter distance than original merging point can be selected as the new merging point. Usually, point with shortest distance would be selected. If there is no point with shorter distance, the original point is selected as the merging point. If the traversal just can find one merging point, the point would be selected. The main idea of the algorithm is to choose a nearest target streaming.

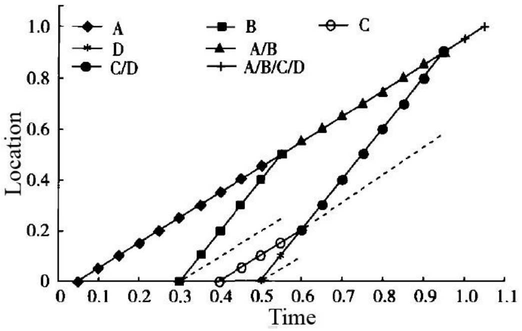

Figure 4. Optimal stream merging

Dynamic program gives a minimum consumption merging tree of all requests. As shows in Figure 4, A, B, C and D represent access of A, B, C, and D respectively; A/B represents merging streaming of access A and access B while C/D represents merging streaming of access C and access D; A/B/C/D represents the merging stream of access A, B, C and D. There is no doubt that it is the best merging method if the consumption is the unique index.

When access C arrives, dynamic program merges access C and B at point 0.5, and merging stream of access B and C merges at point 0.75 with access A. With access D arrives and dynamic program, access D and C merge at point 0.6 and access B and A merge at point 0.55, A and B merging stream and C and D merging stream merge in 0.95. This is the best merging tree and the consumption value is 1.9.

In Figure 5, if access D arrives at point 0.55 and with algorithm ERMT (earliest reachable merge target), access B and C merge at point 0.25, stream B and C merge with access D at point 0.8, stream of B, C and D merges with A at point 1.05. The total consumption is 2.1; with algorithm of CRMT, access of B and C merge at point 0.5, stream of B and C merges with access A at point 0.75, stream of A, B and C merges with access D at point 1.05. The total cost is 2.05. So, CMRT algorithm has better performance than that of EMRT. Of course, both the two algorithms are compatible for the actual application.

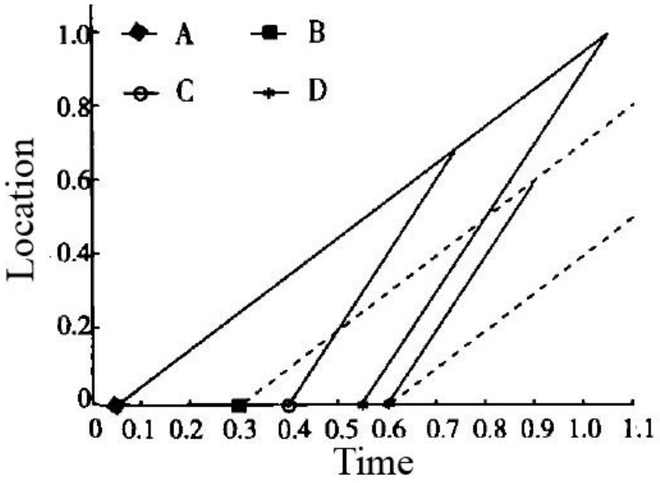

Figure 5. Compatible reachable merge target

Using x_1 , x_2 , x_3 , x_4 represent arriving time of access A, B, C and D. Using dimensionless time unit “1” as time length of video, CA represents the merging point of access A and C, $\overline{CA}$ represents distance between arriving time of access C and CA . Then,

$$CA = x_1 + 2(x_3 - x_1) = 2x_3 - x_1 \tag{3}$$

$$\overline{CA} = x_3 - x_1 \tag{4}$$

Whether target stream B merges with stream D or not, stream D can always merge with stream A. So, if merging point of stream B and D has short distance than that of B and A, streaming D is selected to be the merging point, or else, A is the merging point, that is to say that merging behavior has no change. If streaming D is chosen to be the merging point, difference between two merging methods is the consumption saving. If stream B keep the original merging target of stream A, the total cost would keep constant.

In Figure 5, if access D arrives at time point of 0.6, there is just one target stream B. merging tree of stream B and D would have a less consumption than merging of stream B/C first and then merging B/C with A. The difference between the two merging method can be described as the following:

$$(1-2(x_3-x_2))-(\overline{CA}-\overline{CB})-(1-\overline{DB}) = x_4-(2x_3-x_1) = x_4-CA \quad (5)$$

We know the arriving time of D is earlier than the merging point of stream of B/C and access A. Or else, B can't be the target stream. So value of equation 5 won't be bigger than 0.

From the description above, we can see that CRMT algorithm is more compatible than ERMT to produce merging tree.

Merging tree of CRMT is shown in figure 6. 'a' in the figure represents prefix. When access A arrives at time 0, prefix of streaming multimedia would be get from prefix proxy, and left content acquire from postfix multicast channel. When access B is at the end of prefix, stream B would merge with A at point 0.55; when access C is at the end of prefix, it would seek forward and find two target stream B and A. Due to the appearance of stream C, it can merge with B at point 0.5. So, it merges with B instead of merging of A and B at point 0.55. As the same method, access D merges with forward target stream too.

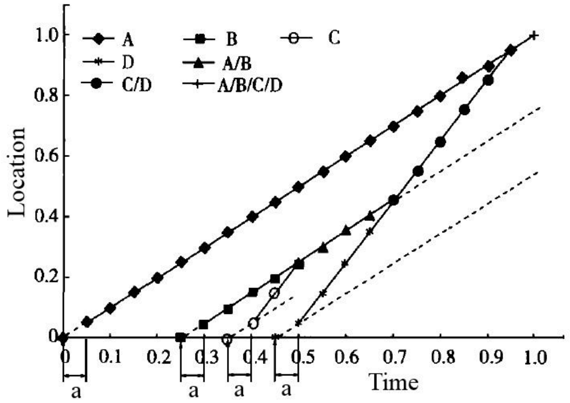

Figure 6. Compatible reachable merge target

Now, there would be four kind of channel in the system: (1) unicast channel between terminal computer and proxy to receive prefix; (2) unicast channel between proxy and server to receive postfix; (3) multicast channel between proxy and terminal computer to play postfix; (4) multicast channel to monitor merging target of proxy postfix.

4. Simulation and Results

The simulation results would be compared with three forms of earliest streaming merge target.

Figure 7 shows the performance comparison with optimal stream merging. The Y axis represents the increasing percentage of server bandwidth compared with optimal stream merging, while X axis represents velocity of terminal computer request in the whole process of video playing. From figure we can see that results of CRMT algorithm is better than that of

ERMT algorithm, and performance difference between CRMT and optimal stream merging is quite little.

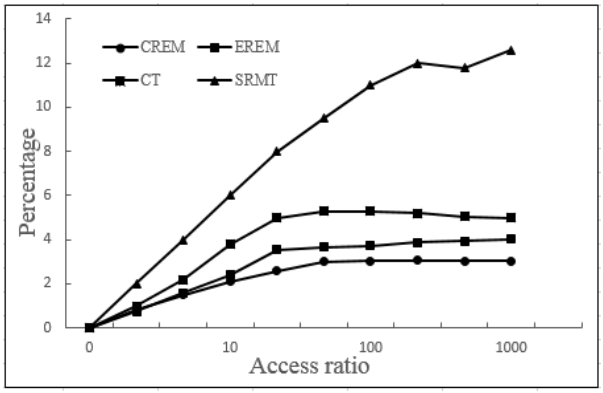

Figure 7. Performance comparison with optimal stream merging

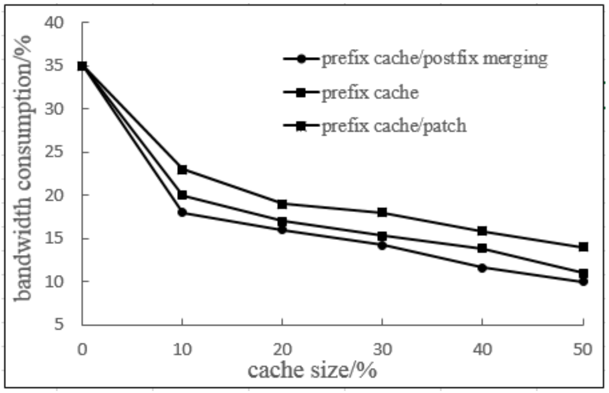

Figure 8. Server bandwidth deduction ratio

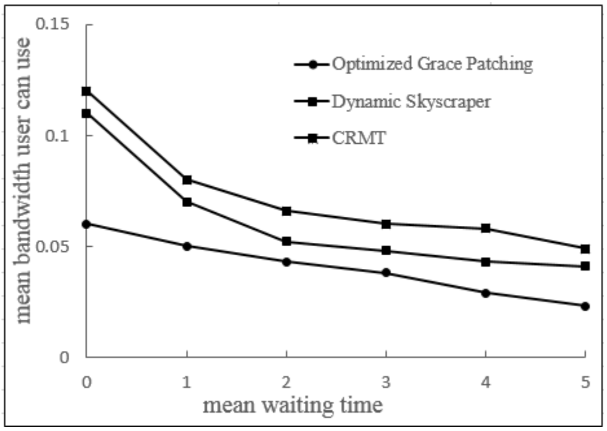

Figure 9. Mean client waiting time

Figure 8 shows that bandwidth cost with server bandwidth deduction ratio. In different size of cache and compare with prefix algorithm, prefix integrated with CRMT takes less bandwidth.

Cache size represents the ratio of size of proxy cache compared with total size of all the multimedia; bandwidth deduction ratio represents the occupied ratio of bandwidth between server and proxy.

Figure 9 shows mean waiting time of terminal computers with different algorithms in fixed server compatible bandwidth. Dynamic adjusting algorithm the paper proposed is mainly to improve the utilizing efficiency, at the same time, it also can decrease the bandwidth.

5. Conclusion

Multimedia is more and more widely used in education, no matter in college classes or senior school class or others places. Number of multimedia education files would be very large in the future. So, it is quite important to study the dispatching method to meet the demand of normal teaching activities.

In the proxy cache based streaming multimedia transmission system, cache strategy and transmission scheduling scheme are the core of the system design. Both algorithm and system can be used in education system. In order to further improve the cache efficiency, saving network bandwidth and improve capacity of the server, paper proposed a prefix cache, postfix multicast/ multicast channel merging algorithm. According to results of the simulation, the algorithm can get better effect on saving bandwidth and decrease start time delay. It also can dynamically adjust the capacity of the segment and cache. At a certain degree, it saves the resource in the network.

Streaming merging method used in proxy cache system is still need further study, such as seeking merging method and combination with other method, and so on.

References

[1] P. R. Voumard, "Multi-media education - student and staff issues", IEEE International Conference on Multi-Media Engineering Education – Proceedings, **(1996)** July 3-5, Melbourne, Australia, pp. 319-324.

[2] M. Tanaka, "Study on the multi-media education system", Proc 1988 Int Conf Syst Man Cybern, **(1988)**, pp. 934-937.

[3] Y. Hu and W. Gui, "Multi-media Technology in education: where are they come and where are they go?", Proceedings of the 2007 1st International Symposium on Information Technologies and Applications in Education", **(2007)** November 23-25, Kunming, China.

[4] S. Sen, J. Rexford and D. Towsley, "Proxy Prefix caching for multimedia streams", In Proceeding of IEEE INFOCOM, vol. 3, **(1999)**, pp. 1310-1319.

[5] R. Rejaie, M. Handley, H. Yu, *et al.*, "Proxy caching mechanism for multimedia Playback streams in the internet", In Proceedings of International web Caching Workshop, **(1999)** March, San Diego, CA.

[6] K. Wu, P. S. Yu and J. Wolf, "Segment-based proxy caching of multimedia streams", In Proceeding of WWW, **(2001)** May, Hong Kong, China.

[7] R. Rejaie, H. Yu, M. Handely, *et al.*, "Multimedia proxy caching mechanism for quality adaptive streaming applications in the internet", In Proceeding of IEEE IN FOCOM, **(2000)** March Tel-Aviv, Israel.

[8] T. Kim and M. H. Ammar, "A comparison of layering and stream replication video multicast schemes", In Proceedings of ACM NOSSDAV 2001, **(2001)** June, NY.

[9] C. Aggarwal, J. Wolf and P. Yu, "On optimal batching policies for video on demand storage servers", In Proc. IEEE Int. Conf. Multimedia Computing and systems, **(1996)**, pp. 253-258.

[10] K. Hua, Y. Cai and S. Sheu, "Patching: a multicast technique for true video on demand services", In Proc. ACM Multimedia, **(1998)**, pp. 191-200.

[11] C. C. Aggaral, J. L. Wolf and P. S. Yu, "On Optimal piggyback merging policies for video-on-demand systems", Proc. ACM SIGMETRICS Conf., **(1996)** May, Philadelphia, PA.

[12] D. L. Eager, M. K. Vermon and J. Zahorjan, “Minimizing bandwidth requirements for on-demand data delivery”, In Proceedings of the 5-th. International Workshop on Advances in Multimedia Information System (MIS’99), **(1999)**, pp. 80-87.

[13] B. N. Amotz and Ladnerre, “Efficient algorithms for optimal stream merging for media-on-demand”, SISM J. Comput., vol. 33, no. 5, **(2004)**, pp. 1011-1034.

[14] D. Eager, M. Vernon and J. Zahojan, “Optimal and efficient merging schedules for video-on-demand servers”, In Proc. ACM Multimedia, **(1999)**, pp. 199-202.

[15] D. Guan and S. Yu, “A new patching channel schedule scheme for video multicast”, IEEE Transactions on Consumer Electronics, vol. 49, no. 2, **(2003)**, pp. 342-347.

[16] H. Wang and Y. Z. Zhong, “Patching first: A novel channel allocation strategy for video-on-demand systems”, Journal of Tsinghua University, vol. 41, no. 4, **(2001)**, pp. 214-217.

[17] Shi, Zhi, Kuo and C. C. Jay, “Recursive patching for video-on-demand (VOD) systems with limited client buffer constraint”, Proceedings - IEEE International Symposium on Circuits and Systems, **(2002)** May 26-29. Arizona, U.S.A.

[18] R. Rejaie, H. Yu, M. Handley, *et al.*, “Multimedia proxy cache mechanism for quality adaptive streaming applications in the Internet” IEEE INFOCOM 2000, vol. 2, **(2000)**, pp. 980-989.

[19] S. Acharya and B. Smith, “Middleman: A video cache proxy server”, Proc. of NOSSDAV, June 2000. NC, USA: IEEE, **(2000)**, pp. 101-116.

[20] D. Zheng, H. Yu and W. Zheng, “Segment-based adaptive proxy caching of multimedia streams”, Journal of Tsinghua University, vol. 47, no. 7, **(2007)**, pp.1204-1207.

[21] S. Subhabrata, J. Rexford and T. Don, “Proxy prefix caching for multimedia streams”, Proceedings of the 1999 18th Annual Joint Conference of the IEEE Computer and Communications Societie, INFOCOM-99, vol. 3, **(1999)**, pp. 1310-1319.

[22] S. Gruber, J. Rexford and A. Basso, “Protocol considerations for a prefix-caching proxy for multimedia streams”, Computer Networks, vol. 33, **(2000)**, pp. 657–668.

[23] L. Shuru, L. Fangmei and C. Zengyu, “Simulation and analysis of video stream transmission based on decodable frame ratio model in wireless network”, Proceedings of SPIE - The International Society for Optical Engineering. **(2012)** October 6-7. Singapore.

[24] K. A. Hua and S. Simon, “Skyscraper Broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems”, Computer Communication Review, **(1997)**, pp. 89-100.

[25] C. C. Aggarwal, J. L. Wolf, Yu and S. Philip, “Adaptive piggybacking schemes for video-on-demand systems”, Multimedia Tools and Applications, vol. 16, no. 3, **(2002)**, pp. 231-250.

Authors

Zhang Yuhong

She received her bachelor's degree of education in Hebei Normal University of Science & Technology, Qinhuangdao, Hebei. (2001) and master's degree of education in Hebei Normal University, Baoding, Hebei. (2007), Now she is a lecturer in Hebei Normal University of Science & Technology, Qinhuangdao,Hebei. Her major fields of study are education technology, vocational education, and information education.

Gu Liwei

She received her bachelor's degree of education in Hebei Normal University Shijiazhuang, Hebei, (2002) and master's degree of Software Engineering in Beijing University of Technology, Beijing, (2010) , Now she is a lecturer in Hebei Normal University of Science & Technology, Qinhuangdao, Hebei. Her major fields of study are education technology, distance learning, software engineering and education.

Wang Yubin

He received his bachelor's degree of engineering in Yanshan University, Qinhuangdao, Hebei. (2000) and master's degree of engineering in Yanshan University (2010) , Now she is a lecturer in Hebei Normal University of Science & Technology, Qinhuangdao,Hebei. His major fields of study are computer network technology.