A Neighbors-Buffering Based Technique to Provide Scalable Distance Learning Service in a Multicast Environment

Tarik Taleb, Nei Kato, and Yoshiaki Nemoto
Graduate School of Information Sciences
Tohoku University
980-8579, Sendai
JAPAN
{taleb,kato,nemoto}@nemoto.ecei.tohoku.ac.jp

Abstract – **In this paper, we propose an approach to provide a significantly scalable video-on-demand service mainly designed for multicast environments where users are geographically close to each other and may have access to the same video feature at nearly the same time. Distance learning is a suitable candidate for this approach. The basic idea behind our proposed approach is to advantage of appropriate buffering of participants of a particular session to satisfy new requests willing to join the same session. Our approach repeatedly transmits popular video data on staggered channels. If a request comes in between staggered start times, the user joins to the most recently started multicast session and then requests the missing part from a nearby neighbor. Users must have enough buffer space to buffer data between staggered transmissions. We refer to our proposed architecture as Neighbors-Buffering Based Video-on-Demand (NBB-VoD) architecture.**

Analytical results show that our proposal achieves a better performance than the already existing system, Unified-VoD, in terms of both scalability and disk-bandwidth requirements.

I. INTRODUCTION

Interactive video-on-demand services, such as distance learning, are likely to form a significant component of the workload of future network applications. In typical proposals for video-on-demand, customers are serviced individually by being allocated a transmission channel and a set of server resources. To provide such a service, known as True Video-on-demand (True-VoD) [2], the system must reserve a dedicated video channel for each user, which ultimately leads to an expensive to operate, and non-scalable system. Near-VoD Systems make use of multicast technology to simultaneously service many users without overloading either the network or the server resource [3][5]. In these systems, each video data is multicast using a predetermined number of channels. For each channel, the assigned video data is periodically repeated over time and channels transmitting the same video data are offset by a time slot. Videos are available only at the beginning of these slots. A customer making a request after the start of a multicast channel will thus have to wait till the upcoming channel starts transmitting the video data. This introduces a significant start-up delay to the customer, which effectively contradicts the on-demand nature of the service. Unified video-on-demand (Unified-VoD) system unifies the existing True-VoD and Near-VoD systems by integrating unicast with multicast transmissions [1]. In this system, instead of scheduling requests that come in between staggered start times for the upcoming multicast channel, the requests will be immediately served by a unicast stream. By so doing, the system can reduce the start-up delay in a multicast environment. However, since even the most powerful server system will always be resource limited, some requests may be denied when a large number of users issue requests in a short period of time for a certain number of popular items.

In this paper, we further increase the system capacity by reducing the effective request arrival rate to the video server. The basic idea of our proposal is to take advantage of appropriate buffering of participants of a particular session to satisfy the maximum number of new requests, willing to join the same session, instead of using unicast channels. A number of local video servers (LVSs) are distributed in a wide service-area network (WSA). Each of them serves a particular service area and the WSA central server serves a group of these local areas. Popular video data are locally replicated at each LVS and are repeatedly transmitted on staggered channels. If a request comes in between staggered start times, the user joins to the most recently started multicast session and then requests the missing part from a nearby neighbor. Users must have enough buffer space to buffer data between staggered transmissions. The impact of our proposal on the system is to increase system capacity and to make better utilization of available unicast streams. We refer to our proposed architecture as Neighbors-Buffering Based video-on-demand (NBB-VoD) architecture. Due to paper limitation, we are obliged to omit a detailed description of our proposed mechanism. Interested readers are kindly referred to [4] for an operational overview of NBB-VoD.

By storing frequently requested videos on local servers, most client requests can be locally served resulting in reducing the transmission cost and the backbone WSA total bandwidth requirement. NBB-VoD can clearly achieve further reduction in the traffic load and network congestion. When a user accesses the local video server LVS, the service manager attempts to satisfy his request by establishing an *interconnection* between the user and this latter's nearest *neighbor* to transmit one portion of the video data while the rest is delivered to the user through a multicast channel directly from the video server. If a large number of requests are similarly satisfied, then reduction in the backbone LSA bandwidth requirement can be significant. This reduction in the bandwidth requirement yields reduction in the video traffic, which is inherently bursty over both short and long time scales. This reduction yields also an alleviation of the congestion in both the network and the bottleneck due to the local video server.

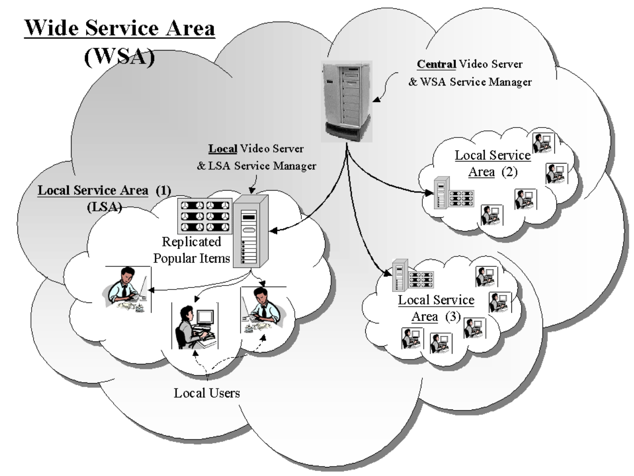

Fig. 1. A two-tiered network architecture: A number of local video servers are distributed in a wide service-area network. Popular video data are replicated at local servers to be locally served while the central server provides the whole area with unpopular items

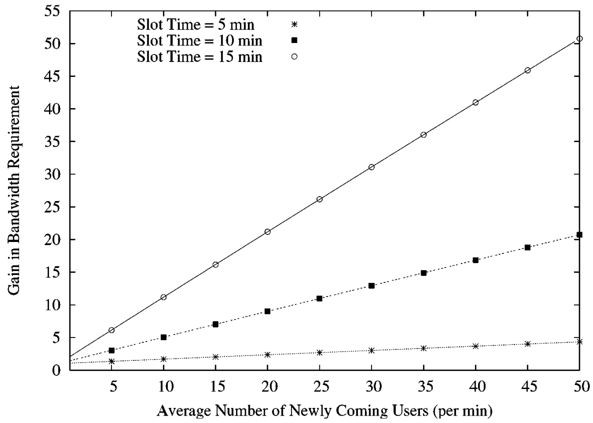

Fig. 2. Performance gain in terms of disk bandwidth requirement: Gain in bandwidth requirement over unified-VoD vs average number of newly coming users per one minute for different values of slot-time.

We believe that NBB-VoD may achieve magnificent performance to provide video-on-demand services to university campus networks where users are geographically close to each other and may desire to have access to the same VoD applications, distance learning, at nearly same time.

The remainder of this paper is organized as follows. In the next section, we describe the major components needed for analytically developing NBB-VoD. Analytical results are discussed in section Ⅲ. We conclude in section Ⅳ.

II. SYSTEM KEY PARAMETERS

Fig. 1 depicts a two-tiered architecture of our proposed system. The service manager uses information about outstanding requests and the availability of resources to accept or reject requests. The video server is responsible for receiving and processing manager signaling and control information. It must retrieve requested data from a variety of storage media while still meeting real-time delivery deadlines. For simplicity of discussion, each WSA is assumed to comprise several local service-areas (LSAs) interconnected by a backbone WSA. We assume that the network connecting video servers and users is multicast capable and provides a sufficiently fast channel for delivering video data. We assume all the local video servers (LVS) and local service managers (LSM) are similar. We thus can focus on only one of them.

We assume that there are N_u unicast channels, N_m multicast channels, and M replicated video data of average length L at each local video server. In our numerical results, we consider the case of 10 popular video data (M=10) and L is set to 90 min. Each video data is assigned a predetermined number of multicast channels, and for each channel, the data is periodically repeated over the service time. Multicast channels transmitting the same video data are offset by a slot time W. It is assumed that all multicast and unicast streams are statistically identical with a transmission capacity C. The unicast channels share the same request queue and serve incoming requests in the first-come-first-served (FCFS) discipline. If we assume that the service time of unicast channels is exponentially distributed, the unicast channels can be then modeled as an M/M/n/n+N queue, where N is the queue capacity.

At the user side, we assume that all users' devices are similar and contain sufficient extra buffering to hold a time slot's worth of frames. The request arrival process is assumed to be Poisson with arrival rate λ. This assumption is appropriate because the number of VoD users is typically large and users generate the service requests independently.

In our proposal, significant gains can be achieved in the case of popular videos. Hence, we focus in our evaluation on only popular data that are replicated at each LVS. We assume that the viewing probabilities of videos follow a normalized geometric distribution. We classify the M popular videos in order of their popularity. The probability that the kth video is selected is given then by:

$$P_k = \frac{\theta^{k-1}(1-\theta)}{(1-\theta^M)} \quad \text{where } k = 1, 2, 3\ldots..,M$$

The parameter θ is called the skew factor.

III. ANALYTICAL RESULTS

A. *Performance Gain in Terms of Disk Bandwidth*

Using the mathematical model developed in [4], we plot the performance gain, in terms of disk bandwidth savings, of NBB-VoD over Unified-VoD versus the average number of newly arriving requests in Fig. 2. Our results show clearly that our proposed scheme significantly outperforms Unified-VOD. As the number of new requests increases, NBB-VoD shows better performance in terms of disk-bandwidth mainly for large values of time-slots, in other words when only few multicast channels are assigned for the video data. This gain can be effectively exploited to improve resources utilization at the local video server.

B. *Performance Gain in Terms of Scalability*

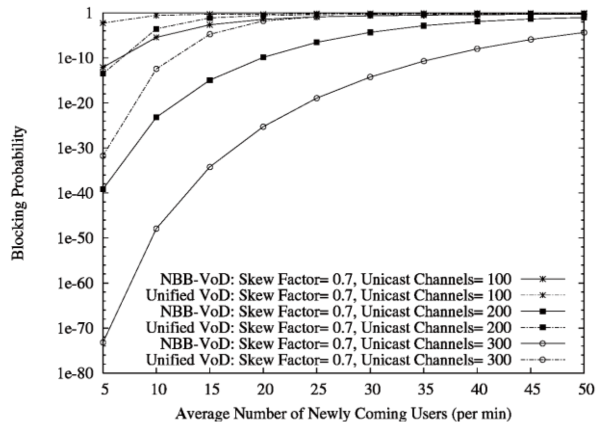

Fig. 3. Performance gain in terms of scalability: requests blocking probability vs average number of newly coming users per min for different number of unicast channels (Skew factor is fixed to 0.7)

We compute numerical results from the M/M/n/n+N queuing model to evaluate the performance of NBB-VoD over Unified-VoD in terms of blocking probability.

At each LVS, we distribute the N_u available unicast channels among the M popular videos in function of their popularities. In other words, for the k^{th} video data, we assign $N_u \cdot P_k$ unicast channels. In our performance evaluation, we mainly focus on the performance of our proposal in case of requests for the first popular video. Similar results are obtained for other popular videos.

Assuming that the requests arrival rate for a particular video in case of Unified-VoD is λ, as NBB-VoD attempts to reduce the number of effective requests to the video server, this arrival rate should be reduced by a factor "c" in case of NBB-VoD. Intuitively, this factor depends largely on the item popularity. In our analytical results, we assume that this factor is equal to the video popularity.

First, we investigate the impact of the number of unicast channels in a LVS on the blocking probability. Fig. 3 illustrates the blocking probability versus the average number of newly coming requests for different number of unicast channels. The skew factor θ is fixed to 0.7. As the number of requests increases, so does the blocking probability in both Unified-VoD and NBB-VoD. It is observed also that the blocking probability decreases as the number of available unicast channels increases. Fig. 3 shows also that the Unified-VoD system has a higher blocking probability than our proposal NBB-VoD as the number of requests increases.

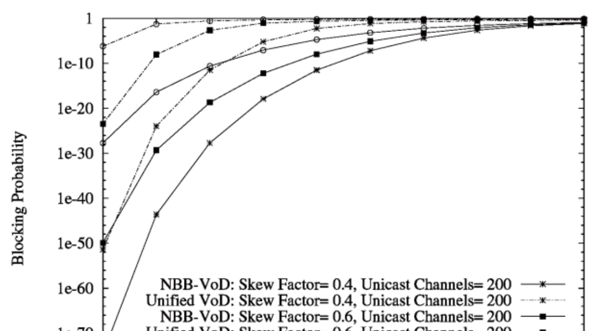

Fig. 4. Performance gain in terms of scalability: requests blocking probability vs average number of newly coming users per min for different values of skew factor (the number of unicast channels is fixed to 200)

To investigate the impact of data popularity on system performance, we plot the blocking probability as a function of the average number of new requests for different values of the skew factor θ in Fig. 4. We fix the number of unicast channels to 200. The results show that the blocking probability increases for larger values of θ in both systems. This increase can be explained in terms of the number of unicast channels assigned to each popular video. For larger values of the skew factor θ, the system assigns more unicast channels for the first popular video data as its popularity increases. When only few number of new requests arrive, the blocking probability increases sharply and more rapidly in case of unified-VoD than in case of NBB-VoD. However, the two systems have similar blocking probability as they approach their capacity, larger values of newly arriving requests.

IV. CONCLUSION

In this paper, we have developed a technique to provide a significantly scalable VoD service. The proposed system is primarily designed for multicast environments where users are geographically close to each other and may desire to view the same video feature at nearly the same time. Our proposed method is believed to achieve magnificent performance to provide distance-learning service to university campus networks where numerous users may have access to the same data.

The importance of our proposal is verified by numerical results, which show that NBB-VoD significantly outperforms Unified-VoD in terms of the disk bandwidth requirement. For different videos with different viewing probabilities, NBB-VoD guarantees lower requests blocking probability than Unified-VoD.

It should be emphasized that there are several implementation issues that must be resolved when applying our proposal to practice. For instance, user's devices should be capable of sending data among themselves in a secure way that prevents illegal intruders from having any unauthorized access. Our proposal could be costly in terms of other resources. For example, as we explore options for optimally making use of user's buffering to increase the system capacity, more network bandwidth and new management software may be required. This will intuitively incur substantial overhead at both the user and system sides. Since our proposal attempts to service a large population of customers, this additional cost can pay for itself in a short time of VoD service utilization.

Leaving these issues aside, we believe that NBB-VoD is a practical method for providing distance learning and represents a major contribution in VoD services area.

V. REFERENCES

[l] J.Y.B. Lee, "UVoD: an unified architecture for video-on-demand services," IEEE Commun. Letters, vol.3, no. 9, Sep. 1999,pp. 277-279.

[2] V.O.K. Li, "Performance model of interactive video-on-demand systems," IEEE J. Select. Areas Commun., vol.14, no. 6, Aug. 1996, pp. 1099-1109.

[3] C.C. Aggarwal and J.L. Wolf, “The maximum factor queue length batching scheme for video-on-demand systems”, IEEE Trans. On Computers, vol.50, no.2, Feb. 2001, pp. 97-110.

[4] T. Taleb, N. Kato, and Y. Nemoto, “Neighbors-buffering based video-on-demand architecture”, accepted for publication in Signal Processing: Image Communication

[5] H.K. Park, "Multicast delivery for interactive video-on-demand service”, in *Proceedings of the 12th International conference on Information Networking, Jan. 1998, Tokyo, Japan, pp. 46-50.*