

Investigating the Performance of Audio/Video Service Architecture I: Single Broker

Ahmet Uyar
Elec. Eng. and Comp. Sci.
Syracuse University
auyar@syr.edu

Geoffrey Fox
Community Grids Labs
Indiana University
gcf@indiana.edu

ABSTRACT

Increasing network bandwidth and computing power provide new opportunities for videoconferencing systems over Internet. The number of homes and small offices with broadband Internet connections are increasing rapidly. Even cell phones will have broadband Internet access in the near future. Therefore, it is not inconceivable to imagine that the trend in the increasing usage of videoconferencing systems will continue by accelerating. This requires universally accessible and scalable videoconferencing systems that can deliver thousands of concurrent audio and video streams. However, developing videoconferencing systems over Internet is a challenging task, since audio and video distribution requires high bandwidth and low latency. Current videoconferencing systems such as IP-Multicast [1] and H.323 [2] can not fully address the problem of scalability and universal accessibility. We propose service oriented architecture for videoconferencing, GlobalMMCS, and use an event brokering middleware, NaradaBrokering, to deliver real-time audio and video streams to high number of users. The performance of the event brokering network is critical to the success of this videoconferencing system. In this paper, we provide comprehensive analysis of a NaradaBrokering broker in the context of audio/video delivery. The results provide guidelines for the deployment of GlobalMMCS in particular, and they provide useful insights for the feasibility of using software based audio/video delivery systems in general.

Keywords: Architectures and Design of Collaborative Systems, Grid-based Collaborative Environments, videoconferencing, distributed event brokers.

1. INTRODUCTION

The availability of increasing network bandwidth and computing power provides new opportunities for distant communications and collaborations over Internet. On one

hand, the number of homes and small offices with broadband Internet connections are increasing rapidly. Even cell phones will have broadband Internet access in the near future with the deployment of 3G standards. On the other hand, the usage of webcams, video camera enabled PDAs, and cell phones are growing by many millions every year. Therefore, it is not inconceivable to imagine that the trend in the increasing usage of videoconferencing systems will continue by accelerating. This will require universally accessible and scalable videoconferencing systems that can deliver thousands or tens of thousands of concurrent audio and video streams. In addition to audio and video delivery, such systems should also provide scalable media processing services such as transcoding, audio mixing, video merging, etc. to support increasingly diverse set of clients.

However, developing videoconferencing systems over Internet is a challenging task, since audio and video distribution requires high bandwidth and low latency. In addition, the processing of audio and video streams is computing intensive. Therefore, it is particularly difficult to develop scalable systems that support high number of concurrent users with diverse set of features. Videoconferencing systems such as IP-Multicast [1] and H.323 [2] can not fully address these problems. These systems focus on delivering the best performance and lack flexible service oriented architecture. IP-Multicast is not universally accessible, and H.323 based systems are not flexible to add new resources and services. We believe that with the advancements in computing power and network bandwidth, more flexible and service oriented systems should be developed to manage audio and video conferencing systems. Therefore, we proposed service-oriented architecture to develop a scalable videoconferencing system, GlobalMMCS [3, 4, 5], based on a publish/subscribe event brokering network, NaradaBrokering [6, 7, 8].

There are two main design principles of GlobalMMCS architecture. First one is to design independently scalable and distributed components for each task performed in videoconferencing systems. Second one is managing the interactions among these components using the principles of service-oriented computing to provide a flexible and dynamic framework to add new computing power and services. We identified that there are three main tasks performed in videoconferencing systems on server side: audio/video distribution, media processing and meeting management. Contrary to conventional videoconferencing systems, we use a distributed event brokering system to deliver all media and data content. This has many advantages, as we pointed out in [9]. Some of these advantages are scalability, support for multiple transport protocols, traversing through firewalls, performance monitoring, and security services provided by NaradaBrokering. Media processing is handled by media servers that are attached to this distribution network. They can scale to arbitrary sizes and they can be distributed in geographically distant locations when necessary. However, the performance of the event brokering network is critical to the success of this videoconferencing system. Our initial tests [9] suggested that this is a viable approach. But, we still needed a comprehensive analysis of the performance of the brokering network. Therefore, in this paper, we provide comprehensive analysis of a NaradaBrokering broker in the context of audio/video delivery. Since the building blocks of the distributed brokering network are brokers, it is essential to know thoroughly the capacity and the limits of a single broker. This helps us to predict the performance of the broker network in distributed settings and to identify the bottlenecks and problems in multi broker environments. In addition, it provides guidelines for the deployment of this videoconferencing system, particularly for small scale organizations.

First, we point out the characteristics of audio and video streams in general and explain the audio and video streams that we used in our tests. In section 3, we determine the quality assessment criteria for audio and video stream delivery in videoconferencing sessions. In the following section, we evaluate the performance of a broker for both single large size and multiple smaller size meetings.

2. CHARACTERISTICS OF AUDIO AND VIDEO STREAMS

The characteristics of audio and video streams are significantly different, though with some similarities. First of all, video streams tend to be much more bandwidth intensive. Secondly, most audio codecs send periodic packages with fixed size data during a session. The only

exception occurs when there is a silence. In that case, no audio packages are sent if the encoder is suppressing the silence. Similarly, video codecs also encode each frame of the stream periodically according to the frame rate of the video stream. Nonetheless, they may generate multiple packages with differing sizes for each video frame according to the changes in the picture between consecutive frames. If there are more changes, more packages are generated for a frame. If there is less action, fewer packages are generated. Moreover, video codecs occasionally send full picture updates. These full updates generate much more packages than regular frame encodings in which only the changes are encoded from the previous frame. Therefore, neither the size of the video packages nor the number of video packages per frame is constant.

We have recorded an audio and a video stream for 2 minutes to use in these tests. The audio stream was 64 kbps ULAW. The audio codec sent an RTP package every 30 millisecond, each 252 bytes long. There were 4100 packages in total without any silence period in the stream. This is a telephone quality audio and widely used in videoconferencing sessions over Internet.

We recorded the video stream of a speaking participant in a video conferencing session setting. This was an H.263 stream with 15 frames per second. The video codec encoded a frame every 66ms. Although the average bandwidth was 280 kbps, the bandwidth was fluctuating mostly between 250 kbps to 310 kbps. 1800 video frames are transmitted during 2 minutes, which had 5610 packages in total. The video codec was dividing the frames that have more than 1 KB of data into multiple packages. The average length of the video packages was 740 bytes. The video codec sent one full picture update frame every 60 frames or every 4 seconds. These frames had much more packages than regular frames, as it can be seen at Figure 1. It shows the number of packages per frame. Although, a frame had 3.1 packages on the average, the number of packages for full updates was changing between 10 to 18.

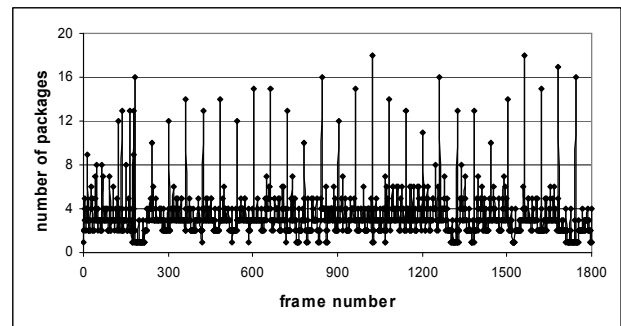


Figure 1 Number of video packages per frame

3. QUALITY ASSESSMENT OF MEDIA DELIVERY

There are three factors that affect the quality of audio and video communication over Internet: the latency, the jitter and the loss rates of packages [10]. We evaluate the quality of the audio and video delivery based on these three factors.

ITU [11] recommends that the mouth-to-ear transit delay of one-way audio should not exceed 400ms to provide acceptable quality. It also points out that the delays up to 300ms satisfy almost all users by delivering good quality. Though, 150ms is preferred to provide excellent quality. On the other hand, there is no agreed upon standard for the transmission delay of video. Nonetheless, in video conferencing applications, video is used along with audio and it should have a similar latency to be in synchrony with audio. Therefore, we require the same latency values both for audio and video.

The total transit delay is the combination of a number of factors: (1) the processing time on the sender machine, (2) the transit time from the transmitter to the broker, (3) the routing time at the broker, (4) the transit time from the broker to the receiver, and (5) the processing time on the receiver machine. In our tests, we tried to minimize all four factors except the routing time at the broker to be able to measure the performance of the broker accurately. We ran all clients and the broker on a gigabit subnet to minimize the transmission latencies. We also eliminated the processing time on sender and receiver machines by recording the times of the packages on the transmitter machine after the encoding of the packages, and on the receiver machine before the decoding. Since the total latency should not exceed 400ms for acceptable quality and 300ms for good quality, we require that a broker should not add more than 100ms of latency. This allows the remaining two transit delays and the processing on receiver and transmitter machines to introduce up to 200ms for good quality and up to 300ms for acceptable quality. We label those packages with higher latency values than 100ms as late packages and consider them as lost packages when assessing the quality of the transmission.

Jitter [12] is defined as the variation in the arrival times of the packages at the receiving end. Since Internet neither provides guaranteed package delivery nor constant delay, packages may take different times to get to the destinations. To smooth out these variations in the arrival of packages, receiving ends implement a playout buffer algorithm. These algorithms delay the playing of packages that arrive earlier to be able to compensate the late ones. However, they discard packages that arrive too

late without playing to the user. Although there is no formal recommendation about the jitter, [13] recommends that the jitter values below 20 ms provide good quality. The jitter values between 20- 50 ms provide acceptable quality and the jitter values beyond 50 ms, results in poor quality. Therefore, in our tests, we require the broker network to introduce less than 10ms of jitter, so that the transmission links can introduce more.

There are two factors that contribute to the package losses experienced by the receiving end. First one is the losses during the transmission. Second one is the losses due to the late arriving packages. In our tests, there were no package losses because of the transmission. It is suggested in [14] that the loss rates should be less than 1.0% for good quality, 1-2.5% for acceptable quality, 2.5-5.0% for poor quality, and 5.0-12.0% for very poor quality. Therefore, in our tests, we tolerate only around 1.0% of package losses.

4. PERFORMANCE ANALYSIS OF A NARADABROKERING BROKER

We conducted extensive tests to evaluate the performance and the limits of a single NaradaBrokering broker. We tested two cases thoroughly: single large scale meetings and multiple smaller scale meetings. We conducted all performance tests on an 8 node Linux cluster with a gigabit network switch among them. All nodes had Double Intel Xeon 2.4GHz CPUs and 2GB of memory. Both NaradaBrokering software and our applications are developed in Java. Therefore, all components of the tests were running as Java applications by using JDK 1.4.2.

In our tests, each meeting is designed as a single speaker meeting with one speaker and many listeners. The speaker publishes an audio or a video stream to a topic on the broker and listeners receive that stream by subscribing to that topic. Each meeting has a dedicated topic. In the case of audio and video combined meetings, audio and video meetings are completely independent. Both of them are seen as separate meetings by the broker network. The multiple speaker meetings are equivalent to multiple single speaker meetings as long as the delivery of streams is concerned by the broker network. Having a meeting of N participants with M speakers are equal to having M single speaker meetings each having N participants. In both cases, M streams are delivered to N clients. Therefore, we have not tested multiple speaker meetings.

4.1. Single Meeting Tests

We tested the performance and the scalability of a broker for three types of single meetings: single audio meetings, single video meetings, and audio and video combined

meetings. We started from small size meetings and went up to large size meetings until the broker can not deliver an acceptable performance. Since the characteristics of audio and video streams are significantly different, it is necessary to test each of these cases independently.

We gathered the results from 12 receivers. These are the first 4, middle 4, and last 4 clients, in the order of their subscription to the meeting topic. Since the broker routes packages in the first-come-first-serve basis, first clients get the best services and last clients get the worst services. Although, this algorithm favors the first clients over the later ones, it provides services with minimum jitter to all clients. For each meeting, the transmitter and measuring receivers were running in the same machine to avoid the clock synchronization problems when calculating the latencies. We calculated the latency and jitter value of every package transmitted using millisecond resolution. Jitter values are calculated according to the formula given in RTP specification [12]. When calculating the latencies, we ignored the first 100 packages to compensate for start up costs.

In online meetings the receiving participants send RTCP packages to report to other participants and to the transmitter about their presence in the meeting and the quality of service they are receiving. In small size meetings, the extra load of RTCP packages is negligible. But in large scale meetings, the load of RTCP packages can become cumbersome, since each RTCP package is delivered to every other participant in the meeting. To avoid this, RTP specification [12] requires scaling down the RTCP package sending interval when the number of participants in a session increases. It recommends that RTCP packages should use %5 of the total session bandwidth. However, our tests showed that JMF library does not scale down the RTCP package sending intervals in large size meetings. The distribution of RTCP packages dominated the large size meetings. Therefore, we have disabled RTCP package delivery in our tests. However, this should not affect the performance tests of the broker significantly, since RTCP packages must use at most %5 of the session bandwidth.

4.2.1. Single Audio Meeting Tests

We tested single audio meetings with varying number of participants on a broker. Table 1 shows the summary of the results of these tests. Each row in the table shows a test case with the given number of participants in the meeting at the first column. Second column shows the average latencies of the first subscribed user and the third column shows the average latencies of the last user. Fourth column shows the average latencies of 12 participants from which we gather the results. Fifth column shows the average jitter values calculated for the

average latencies of 12 participants. Sixth column shows the percentages of packages that arrive later than 100ms for all measuring clients.

Since all audio packages are the same size and they are evenly distributed on time (a package every 30ms), the routing of all packages in a stream takes the same amount of time on the broker. As long as the routing of a package takes less than 30ms, the routing of one package in the stream does not affect the routing of the next one. Therefore, the latency values of consecutive packages are almost the same for a user in a meeting. This results in very small jitter values. Moreover, the latency values for the first client are always constant and very small until the broker is overloaded. In addition, the routing time of a package increases linearly by the number of participants in the meeting. This can be seen at the latency values of the last user in the meeting at the table.

The broker is overloaded when the routing of a package takes more than 30ms. In that case, the next package arrives before the routing of the current package is completed. Each package delays the routing of the next one and the latency increases constantly for the upcoming packages in the stream. In this test, the broker is overloaded when there are 1600 participants in the meeting.

Table 1 Test results for single audio meetings

Total Users	L(1) (ms)	L(N) (ms)	L(a) (ms)	J(a) (ms)	A(a) (%)
12	0.5	0.7	0.6	0.18	0
100	0.5	2.3	1.4	0.15	0
400	0.5	7.9	4.2	0.21	0
800	0.5	15.5	8	0.18	0
1200	0.5	22.6	11.6	0.22	0
1400	0.5	26.5	13.5	0.26	0
1500	3.3	32.3	17.8	0.44	0.2
1600	2260	2290	2275	1.2	100

In summary, the broker can support up to 1400 participants in one audio meeting by providing excellent quality audio delivery with very small latency and jitter values. When there are 1500 participants, some packages starts arriving late, and the broker is close to being overloaded.

4.2.2. Single Video Meeting Tests

Table 2 shows the results for single video meeting tests. The columns of Table 2 are the same as the columns of Table 1. The only difference is the column 6 that has the results for late arriving packages of the last subscriber in video meetings.

Contrary to audio meeting test results, in this case, the latency and jitter values increase rapidly. In addition, some packages start arriving late when there are 400 participants in the video meeting long before the broker is overloaded. The main reason for this is the uneven distribution of packages in the video stream throughout the transmission. Since, there are multiple packages on each video frame and they are published one after another on the broker, later packages in the frame wait the earlier ones to be routed at the broker. This results in more increases on the latency depending on the number of packages on a frame. Particularly the latency of packages in full picture update frames increases much faster. Figure 2 shows the latency values of all 5610 packages transmitted for the last receiver in the video meeting with 400 receivers. The peaks on the latency graph correspond to the full picture update packages. Although, the broker is overloaded when there are 900 participants, the quality of video stream delivery becomes unacceptable when there are 500 participants, because of the high rate of late arriving packages. In summary, single video meetings utilize the broker resources poorly and the uneven distribution of packages in video streams results in late package arrivals long before the broker is overloaded. This limits the number of supported participants in single video meetings significantly.

Table 2 Test results for single video meetings

Total users	L(1) (ms)	L(N) (ms)	L(a) (ms)	J(a) (ms)	A(n) (%)
12	1.0	1.3	1.2	0.44	0
100	3.1	5.0	4.0	2.0	0
300	10.2	16.2	13.2	7.8	0
400	13.4	21.2	17.3	10.1	0.7
500	18.2	28.5	23.4	13.2	3.0
700	29.8	43.7	36.8	18.1	8.4
900	93.7	111	102	23.8	40

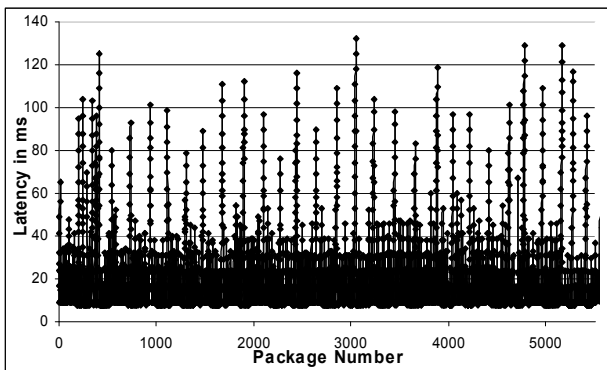


Figure 2 Latency values of the last receiver in the video meeting with 400 participants

We should also note that in single audio meeting tests, the broker is overloaded when its outgoing bandwidth is 102 Mbps, and in single video meeting tests it is overloaded when its outgoing bandwidth is 278 Mbps. The reason for this big difference is the package sizes of audio and video streams. While the sizes of audio packages are 252 bytes, the average video package size is 752 bytes. This shows that it takes very similar amount of time for the broker to route these two types of messages. The total number of packages is almost the same at the point of saturation in both cases.

4.2.3. Audio and Video Combined Meeting Tests

Contrary to the pervious two cases, in this test there are two concurrent meetings. The delivery of one stream affects the delivery of the other. When we performed some initial audio and video combined meeting tests, we observed that the delivery of video streams affected the delivery of audio streams significantly. There was only one queue at the broker for all packages and the routing algorithm routed the packages in a first-come-first-serve basis. Therefore, audio packages needed to wait for the video packages. Although, the audio packages are evenly distributed on time and the latency of consecutive packages for a user is the same in a meeting, when an audio meeting is held together with a video meeting, the latency graph of the audio meeting resembled the latency graph of the video meeting at Figure 2. This increased both the latency and the jitter for audio packages significantly. This was unacceptable, since the audio communication is the fundamental part of a video conferencing session while the video communication is an optional feature. Although having the video feed of the remote party improves the quality of the communication, it is much more important to have a smooth and uninterrupted voice communication [15]. Therefore, we modified the routing algorithm at the broker and introduced another queue for audio packages. We have given priority to audio package routing over all other messages at the broker. When an audio package arrives at the broker, the broker routes this audio package first as long as it is done routing the current package. Therefore, it minimizes the routing times of audio packages. While this approach ensured the quality of audio delivery, it did not reduce the quality of video communications significantly, as the following tests demonstrate.

Table 3 shows the results gathered from the audio meeting when there are one audio and one video meeting at the broker. The columns are the same as the columns of Table 1 in single audio meeting tests. The average latency values for audio and video combined meetings are only slightly higher than the average latency values of single audio meetings. When there are 600 participants, there is only 5ms difference. Therefore, the impact of the video

meeting is very limited on the performance of the audio meeting, because of the priority given to audio package routing at the broker.

Table 4 shows the results gathered from video meetings when there are one audio and one video meetings at the broker. All columns are the same as the columns of Table 3 of the single video meeting tests. Figure 3 shows the average latency values of the video meeting participants at audio and video combined meetings, and the average latency values of the single video meeting participants for comparison. Although the latency values of video meeting participants in audio and video combined meetings are higher than the average latency values of single video meeting participants, the difference is very small until the broker is overloaded in audio and video combined meeting tests. When there are 400 participants, the difference is only 5ms. In addition, there are no late arriving packages for 300 participants in audio and video combined meetings and there is %1.3 late arriving packages for the last user for 400 participants. This value is slightly higher than the single video meeting test in which the late arrival rate was %0.7 for 400 participants. Therefore, one broker can support up to 400 participants in audio and video combined meetings. This shows that the impact of an audio meeting is very small on the latency of video meeting participants. In audio and video combined meetings, the broker supports almost the same number of participants as in the case of single video meetings. The main reason for this is the better utilization of the broker when there are two concurrent meetings.

Table 3 Audio results from audio + video meetings

Total Users	L(1) (ms)	L(N) (ms)	L(a) (ms)	J(a) (ms)	A(a)
12	0.7	0.9	0.8	0.3	0
100	0.9	2.6	1.8	0.73	0
300	2	7.5	4.8	2.67	0
500	4.4	13.4	8.9	3.84	0
600	5.8	16.7	11.2	5.46	0

Table 4 Video results from audio + video meetings

Total users	L(1) (ms)	L(N) (ms)	L(a) (ms)	J(a) (ms)	A(N) (%)
12	1.3	1.5	1.4	0.5	0
100	3.7	5.7	4.7	2.1	0
300	11.3	17.2	14.2	7.5	0
400	18.1	26.1	22.1	10.9	1.3
500	26.7	36.8	31.8	13.5	5.5
600	169	181	175	16.2	60.0

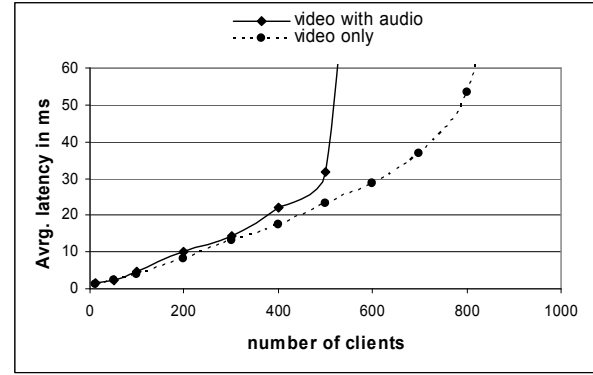


Figure 3 Comparison of average latencies of audio+video meetings and single video meetings

4.2. Multiple Meeting Tests

Similar to single meeting tests, we tested the performance and the scalability of a broker for three types of multiple concurrent meetings: multiple audio meetings, multiple video meetings, and multiple audio and video combined meetings. However, we do not report the multiple audio meeting test results because of the space limitations. We had 20 participants in all meetings. We gathered the results from 10 meetings. These meetings had been chosen in no particular order, since each meeting was independent of others.

4.2.1. Multiple Video Meeting Tests

The results of the multiple video meeting tests are summarized at Table 5. The first column shows the number of meetings on each test on the broker. The next three columns show the averages of the latency, jitter and late arrivals of 10 receivers from 10 different meetings. The next column shows the amount of total data coming to the broker and the last column shows the total amount of data the broker is sending out to receivers.

Table 5 Multiple video meetings tests, each meeting having 20 users

M	L(a) (ms)	J(a) (ms)	A(a) (%)	In BW (Mbps)	Out BW (Mbps)
5	2.25	0.68	0	1.43	28.7
15	3.17	0.86	0	4.30	86.1
25	5.94	1.30	0	7.17	143.5
30	6.80	1.37	0	8.61	172.2
35	10.6	1.52	0.7	10.04	200.9
40	81.1	1.80	19	11.48	229.6

Figure 4 shows the average latency values of multiple video meeting tests and single video meeting tests for the same number of total participants. This graph shows that the average latency values of the multiple video meeting participants are much smaller than the average latency

values of the single video meeting tests until the broker is overloaded. Similarly, the average jitter values of this test are much smaller than the average jitter values of the single video meeting tests. Consequently, the percentages of late arriving packages are much lower. Now, there are no late arriving packages for 30 video meetings with 600 participants, and there are very few late arriving packages for 35 meetings with 700 participants. Therefore, the broker can support 30 video meetings with 600 participants with excellent quality. The main reason for this big difference is the better utilization of the broker when there are multiple video meetings. Since the packages of each meeting arrive randomly distributed on time, they utilize the broker CPU and bandwidth more efficiently. In the single video meeting case, while the broker sits idle when there is no package, it is occupied for a long time when a package arrives. In addition, smaller meeting size in multiple meeting tests ensures that the latency values do not have peaks. The routing of multiple packages in a video frame does not result in late arriving packages since they are routed to only 20 participants. Figure 5 shows the latency values for all video packages transmitted for a participant in a meeting when there are 30 video meetings on the broker. There are no peak values for latencies of full picture update frames as in the case of single video meetings at Figure 2. In summary, when there are multiple meetings, the broker supports much higher number of participants providing better quality services with very small latency and jitter values.

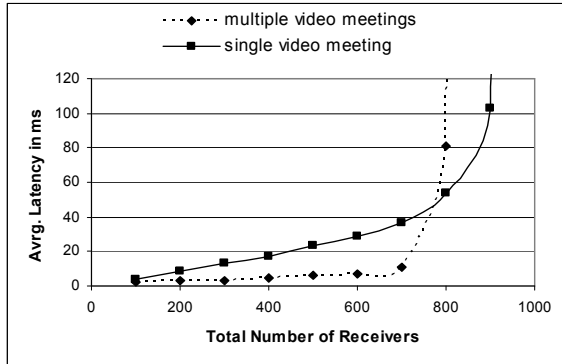


Figure 4 Comparison of average latencies of single and multiple video meetings

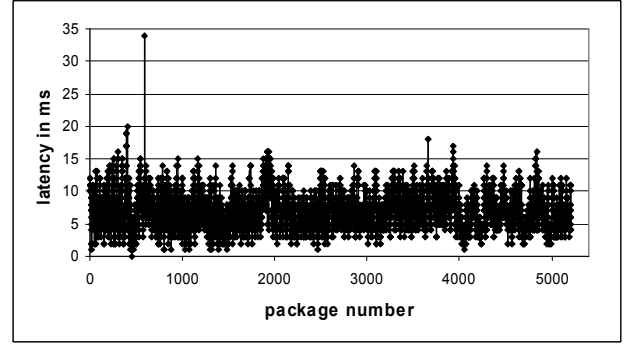


Figure 5 Latency values of video packages for 30 video meetings with 600 participants in total

4.2.2. Multiple Combined Meeting Tests

In the case of multiple concurrent audio and video meetings, we had the same number of audio and video meetings on the broker with 20 participants each. We gathered two sets of results. Table 6 shows the results gathered from audio meetings and Table 7 shows the results gathered from video meetings. First columns show the total number of concurrent meetings. Half of them are audio meetings and the other half are video meetings. Similarly, the second column shows the total number of participants in these meetings. Half of them are audio meeting participants and the other half are video meeting participants. The remaining columns are the same as the columns of Table 5.

Table 6 Audio results from audio + video multiple meeting tests

M	L (a) (ms)	J (a) (ms)	A (a) (%)	In BW (Mbps)	Out BW (Mbps)
10	1.7	0.5	0	1.75	35.1
20	2.5	0.9	0	3.51	70.2
30	3.3	2	0	5.26	105.3
40	4.9	2.2	0	7.02	140.4
50	46.8	2.8	16	8.77	175.5
60	9287	6.6	100	10.53	210.6

Similar to the previous multiple video meeting tests, until the broker is overloaded around 50 meetings with 1000 participants, the average latency is very small for both audio and video participants. In addition, there are no late arriving packages until 800 participants. These results show that 40 meetings (20 audio and 20 video meetings) can be conducted simultaneously on this broker with excellent quality. Furthermore, these results show that the routing of audio streams does not degrade the performance of video streams significantly. Although audio latencies are better than video latencies, both provide very low latency for videoconferencing.

Table 7 Video results from audio + video multiple meeting tests

M	L (a) (ms)	J (a) (ms)	A (a) (%)	In BW (Mbps)	Out BW (Mbps)
10	2	0.7	0	1.755	35.1
20	2.62	0.85	0	3.51	70.2
30	5.25	1.3	0	5.265	105.3
40	6.5	1.56	0	7.02	140.4
50	76.2	1.96	23	8.775	175.5
60	9421	4.12	100	10.53	210.6

5. CONCLUSION

In this paper, we investigated the performance and the scalability of a NaradaBrokering broker to conduct real-time audio and video meetings. We evaluated the performance of the broker for both single large scale meetings and multiple smaller scale meetings. These tests showed that a broker can support up to 1500 users in a single audio meeting and up to 400 users in a single video meeting. They also demonstrated that when there are one audio and one video meeting concurrently, up to 400 participants can be supported in both meetings. The main reason for this is the better utilization of broker resources when there are two meetings, compared to having only one video meeting. The performance tests for multiple smaller size meetings showed that a broker can support 35 video meetings with 700 users in total. The number of supported users in this case is much higher and the quality of the service is much better with smaller latency and jitter values compared to the single video meeting tests. This shows that multiple smaller meetings use the broker resources more efficiently. Finally, the broker supported 20 audio and 20 video meetings, each having 20 users. We should remember that these numbers depend on the underlying machine and the audio and video streams being used. We conducted these tests on a dual processor Linux machine with widely used audio and video formats with typical bandwidths.

In summary, a NaradaBrokering broker can provide audio and video conferencing services to a few hundred users with very good quality. A small or middle size organization can deploy GlobalMMCS videoconferencing system to provide videoconferencing services. Larger organizations need to deploy distributed brokers to support higher number of users in geographically distant locations. We investigate the performance of the brokering system in distributed settings in [16].

6. REFERENCES

[1] K. Almeroth, "The Evolution of Multicast: From the Mbone to Inter-Domain Multicast to Internet2 Deployment", IEEE Network, Jan 2000, Volume 14.

[2] ITU-T Recommendation H.323, "Packet based multimedia communication systems", Geneva, Switzerland, Feb. 1998.

[3] <http://www.globalmmcs.org>

[4] Geoffrey Fox, Wenjun Wu, Ahmet Uyar, Hasan Bulut, Shrideep Pallickara. "Global Multimedia Collaboration System". Concurrency And Computation: Paractice and Experience. 2004; 16:441-447

[5] Ahmet Uyar, Wenjun Wu, Geoffrey Fox. "Service-Oriented Architecture for Building a Scalable Videoconferencing System". Technical Report June 2004.
<http://grids.ucs.indiana.edu/ptliupages/publications/>

[6] <http://www.naradabrokering.org>.

[7] Shrideep Pallickara and Geoffrey Fox. NaradaBrokering: A Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids. Proceedings of ACM/IFIP/USENIX International Middleware Conference Middleware-2003. pp 41-61.

[8] Geoffrey Fox and Shrideep Pallickara. An Event Service to Support Grid Computational Environments. Journal of Concurrency and Computation: Practice & Experience. Special Issue on Grid Computing Environments. Volume 14(13-15) pp 1097-1129.

[9] Ahmet Uyar, Shrideep Pallickara, Geoffrey Fox, "Towards an Architecture for Audio/Video Conferencing in Distributed Brokering Systems", The proceedings of The 2003 International Conference on Communications in Computing, June 23 - 26, Las Vegas, Nevada, USA.

[10] M. Claypool and J. Tanner. "The Effects of Jitter on the Perceptual Quality of Video", In Proceedings of the ACM Multimedia Conference, Vol. 2, Orlando, Florida, USA, November 1999.

[11] ITU-T Recommendation G.114, One Way Transmission Time. (05/2003).

[12] IETF RFC 3550. RTP: A Transport Protocol for Real-Time Applications. July 2003.
<http://www.ietf.org/rfc/rfc3550.txt>.

[13] Prasad Callyam, Mukundan Sridharan, Weiping Mandrawa, Paul Schopis: Performance Measurement and Analysis of H.323 Traffic. PAM 2004: 137-146.

[14] Les Cottrell, Warren Matthews and Connie Logg. Tutorial on Internet Monitoring & PingER at SLAC, <http://www.slac.stanford.edu/comp/net/wan-mon/tutorial.html#loss>

[15] Isaacs, E. and Tang, J. "What video can and can't do for collaboration: A case study." In Proceedings of ACM Multimedia '93. Anaheim: ACM, 1993.

[16] Ahmet Uyar, Geoffrey Fox. Investigating the Performance of Audio/Video Service Architecture II: Broker Network. The 2005 International Symposium on Collaborative Technologies and Systems. May 2005, Missouri, USA.