Investigating the Performance of Audio/Video Service Architecture II: Broker Network

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. In addition to homes and small offices, even the cell phones will have broadband Internet access in the near future. Therefore, we can imagine that the trend in the increasing usage of videoconferencing systems will continue. 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 evaluate the performance of NaradaBrokering broker network in distributed settings in the context of audio/video delivery. The results demonstrate that even small number of brokers can deliver audio/video streams to more than a thousand users with very good quality. They also 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].

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 [8]. 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 analysis of the performance of a single NaradaBrokering broker in the context of audio/video delivery [8] showed that a single broker can support a few hundred participants in both large and small scale meetings with very good quality service. In this paper, we investigate the performance and the scalability of the broker network in distributed settings. This analysis will provide guidelines for the deployment of this videoconferencing system in particular, and they provide useful insights for the feasibility of using software based audio/video delivery systems in general.

First, we explain the scope of the performance tests. In section 3, we evaluate the routing algorithm in NaradaBrokering and propose improvements to provide better scalability. In section 4, we present the single large scale meeting test results in distributed settings. In section 5, we investigate the performance of the broker network for multiple smaller scale meetings. In section 6, we present the test results in wide area networks. In the final section, we conclude by summarizing the results.

2. EVALUATING THE PERFORMANCE OF THE BROKER NETWORK

We investigate the quality of the service provided and the total number of users supported. Distributed broker network provides both opportunities and challenges for providing better quality service and supporting higher number of users. We investigate the performance and the scalability of the broker network for large and small size meetings. We conduct performance tests in controlled settings to eliminate outside factors and measure the parameters that affect the performance of the brokers precisely. These tests demonstrate the capacity of the broker network and provide guidelines to utilize the resources of the broker network efficiently.

We performed all tests using Java. NaradaBrokering software is written in Java. The parts of GlobalMMCS that we use in these tests are also written in Java using Java Media Framework [10]. We recorded a video stream for 2 minutes to use in these tests. The video stream was an H.263 stream with 15 frames per second. It had the average bandwidth of 280 kbps. It was the video stream of a speaking participant in a videoconferencing setting. We give more details about this stream in [9].

We outlined the quality assessment criteria for evaluation of the results of the performance tests in [9]. We require the broker network not to introduce more than 100ms of latency when routing packages. We label packages that take more than 100ms as late arriving packages and consider them as lost packages when assessing the quality of the stream delivery. In addition, we require the loss rates to be less than 1.0%.

3. DELIVERY PRIORITY FOR INTER-BROKER TRAFFIC

A NaradaBrokering broker routes received packages in the following manner. First, every received package is placed into a first-come-first-serve queue. The routing thread picks up the first arrived package from this queue, calculates the destinations, and transmits it to all subscribers of that topic in the order of their subscription. The routing thread continues to route packages one by one as long as there are packages in this queue. In distributed settings, if a package needs to be delivered to other brokers as well, the routing thread first sends the package to other brokers and then to local subscribers. This is to avoid introducing more delays to the transit time of packages that need to travel multiple brokers.

When we started testing the performance of the distributed brokers for a single meeting, we observed that an overloaded broker along the path from the source client to the destination can introduce significant delays to the transmission time of packages. Although other brokers might not be overloaded, their subscribers can still be affected severely by the load of an overloaded broker along the path. In addition, the travel times for packages increases significantly when they go through multiple brokers. Each broker along the path introduces unnecessary delays. This limits the number of brokers a package can travel. Therefore, it limits the scalability of the broker network considerably. We demonstrate this in the following test case.

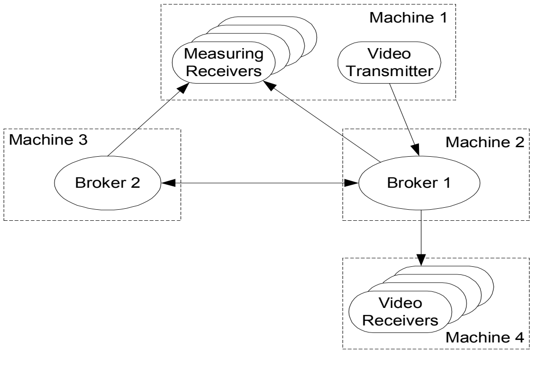

Figure 1 Single video meeting test with two brokers

We set up an NB cluster with two brokers (Figure 1). We initiated one video meeting and loaded the first broker with 400 participants and the second broker with only 6 participants. One video stream is published to the meeting through the first broker. We gathered the results from the first and the last receivers of both brokers. Therefore, they provide the best and the worst results among the meeting participants in both brokers. We performed these tests in a Linux cluster with 8 nodes. Each node had 2.4GHz Dual Intel Xeon CPU and 2 GB of memory.

Table 1 shows the average latencies of 5610 video packages transmitted during the tests. The latency values of both receivers of the second broker are very close to the latency values of the first receiver of the first broker. This test shows that the first broker introduces significant delays to the transit times of video packages transmitted to the second broker. Although the second broker has very few subscribers, its subscribers are still bounded by the first broker. The subscribers of the second broker can not get a service better than the first receiver of the first broker. The main reason for this is the first-come-first-serve queue in the first broker. When there are multiple packages in this queue, the later ones need to wait the earlier ones to be routed to all 400 local subscribers. This can be eliminated by a mechanism that will route packages first to other brokers in the system without waiting local subscribers to be served. This can be implemented by introducing another first-come-first-serve queue and another thread to the broker.

Table 1 Latency values for single video meeting test for two brokers

	First user Latency (ms)	Last user Latency (ms)	Avg. latency (ms)
Broker1	15.83	24.55	20.20
Broker2	16.07	16.18	16.13

3.1. Double Queuing Algorithm

Double queuing algorithm separates inter-broker delivery of packages from local client deliveries. It aims to introduce minimum delays to packages that will be routed to other brokers in the system. In addition to an additional queue, it also introduces another routing thread. Figure 2 depicts this algorithm. Received packages are first placed into the first queue. The first thread picks up a package from this first-come-first-serve queue and delivers it to other brokers in the system if necessary. It hands it over to the second thread by placing it into the second queue after finishing the inter-broker routing. Then it continues to route the next package in the first queue without waiting the second thread to serve the local clients. The second thread continues to serve the local clients as long as there are packages in the second queue. Since these two threads have similar priorities they work concurrently.

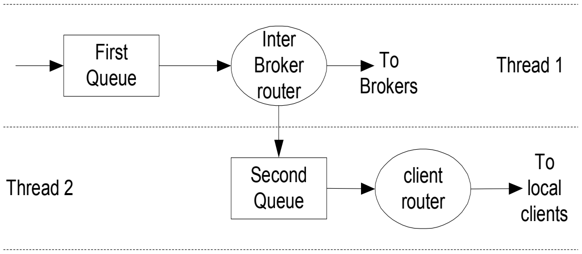

Figure 2: Double queuing algorithm.

We should also remember that there are very few brokers to route a package in a typical distributed NB network. A broker is connected to less than 5 other brokers in most cases. Therefore, inter-broker routing will take very small amount of time compared to the delivery of packages to high number of local clients. Another aspect of this algorithm is that there are two separate queues for both of these queue layers. As it is explained in [9], audio package routing has priority over other packages in brokers. Therefore, there is one audio queue and another for all other packages. The threads first route the audio packages if there is any, then they route the other packages.

We repeated the single video meeting test for two brokers with the same setting for the double queuing algorithm. Table 2 shows the results. As it can be seen, the latency values of the receivers of the second broker are very small. The only overhead introduced by the first broker is the overhead of routing packages to another broker. The high number of clients in the first broker does not impact the performance of the receivers of the second broker. This new algorithm eliminates cases where an overloaded broker severely affects the performance of the subscribers of other brokers.

Table 2 Latency values for single video meeting test with double queuing algorithm for two brokers

	First user Latency (ms)	Last user Latency (ms)	Avg. Latency (ms)
Broker1	16.07	24.92	20.52
Broker2	1.41	1.62	1.52

Another very important advantage of the double queuing algorithm is that it enables the brokering network to grow to high number of brokers. As we can see from the latency values at Table 2, traveling of packages through a relatively loaded broker takes only around 1ms, compared to 15ms previously. Now, even going through 10 brokers for a package only introduces around 10ms of overhead when the transmission delays between the brokers are ignored. This makes it possible for packages to travel many brokers along the way from sources to destinations. Therefore, it enables the broker network to grow in size.

Table 3 Single video meeting test with double queuing for two brokers each having 200 users

	First user Latency (ms)	Last user Latency (ms)	Avg. Latency (ms)
Broker1	7.94	12.37	10.16
Broker2	8.24	12.67	10.45

In addition, the double queuing algorithm enables the distribution of the load of a large size meeting among multiple brokers to provide better quality services with smaller latency and jitter values. Since brokers add minimum delays to packages traveling to other brokers, it makes adding extra brokers into the system very efficient. To demonstrate this, we conducted another test in which each broker had 200 participants. Table 3 shows the results. The latency values of the first broker and the second broker are very similar. The values of the second broker are slightly higher because the packages travel through the first broker to reach to the second broker. When we compare the results of Table 3 and Table 2, we see that the distribution of 400 participants into two brokers reduces the average latencies by fifty percent. This illustrates that more brokers can be introduced into a system to provide better quality service. Similarly, more brokers can be added to provide services to higher number of participants. We show this in the next section.

4. SINGLE MEETING TESTS

Since traveling of packages through multiple brokers adds very small amount of overhead, the number of supported users in a meeting can be increased almost linearly by adding new brokers. In this section, we evaluate the performance of the broker network with four brokers for a single video meeting. Four brokers are connected as in (Figure 3). For the tests in this section, we used two Linux clusters, each having 8 nodes. The nodes of the first cluster had 2.4GHz Dual Intel Xeon CPU and 2 GB of memory. The nodes of the second cluster had 2.8 GHz Dual Intel Xeon CPU and 2GB of memory. Both clusters had gigabit network bandwidth among its nodes.

The receivers were evenly distributed among the brokers and equal number of receivers joined the meeting through each broker. We gathered the results from the last users of every broker. Table 4 shows the average latency values from each broker for the 5610 video packages exchanged. The latency values of broker1 and broker2 are very similar to each other. Similarly, the latency values of broker3 and broker4 are very similar. Last two brokers perform better than the first two brokers; because the second Linux cluster have superior computing power.

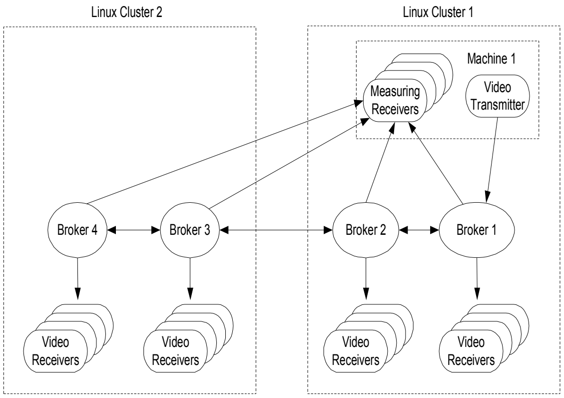

Figure 3 Single video meeting tests with four brokers

As the latency values show, adding new brokers increases the capacity of the broker network as much as the capacity of the added machines. In this case, since all brokers have very similar computing power, each broker increases the capacity of the broker network almost linearly.

Table 5 shows the percentages of late arriving packages. For the first two brokers, the percentage of late arriving packages is %1.9 when there are 400 participants. They can support up to 400 users. For the last two brokers, the rate of late arriving packages are less than %1.0 for the same number of participants. They support 400 users comfortably. In total, four brokers support close to 1600 participants in a single video meeting.

Table 4 Latencies of last users in single video meetings

Users per broker	users in total	B1 (ms)	B2 (ms)	B3 (ms)	B4 (ms)
200	800	12.57	12.93	12.39	12.79
300	1200	18.78	19.25	17.55	18.03
400	1600	25.47	26.02	22.84	23.55
500	2000	32.61	33.72	28.53	29.38
900	3600	685.7	818.9	104.5	90.17

Table 5 Percentages of the late arriving packages for last users in single video meetings

Users per broker	Users in total	B1 (%)	B2 (%)	B3 (%)	B4 (%)
200	800	0	0	0	0
300	1200	0.29	0.27	0	0.02
400	1600	1.87	1.92	0.73	0.73
500	2000	4.01	4.43	2.41	2.4
900	3600	93.5	93.8	37.8	31.3

In summary, these tests demonstrate that the broker network scales well in distributed settings when delivering audio and video streams to high number of participants in large scale meetings. The scalability of the broker network increases almost linearly by the number of brokers. The overhead of going through multiple brokers for a stream is not significant, since inter-broker routing has priority over local client routings.

5. MULTIPLE MEETING TESTS

The behavior of the broker network is more complex when there are multiple concurrent meetings compared to having a single meeting. Having multiple meetings provide both opportunities and challenges. As we have seen in the single broker tests with multiple video meetings [9], conducting multiple concurrent meetings on a broker can increase both the quality of the service and the number of supported users. This can also be achieved in multi broker setting as long as the size of these meetings and the distribution of clients among brokers are managed properly. If the sizes of meetings are very small and the clients in meetings are scattered around the brokers, then the broker network can be utilized poorly. Inter-broker stream delivery can reduce the number of supported users significantly. The best broker utilization is achieved when there are multiple streams coming to a broker and each incoming stream is delivered to many receivers. If all brokers are utilized fully in this fashion, multi broker network provides better services to higher number of participants. To investigate this, we conducted multiple video meeting tests with two different meeting sizes.

We used the same broker organization scheme as the single meeting tests in the previous section. There were four brokers connected as a chain. In this case, all brokers were running in the same Linux cluster that has 8 identical nodes with 2.8 GHz Dual Intel Xeon CPU and 2GB of memory. We also used five other Linux machines that are connected to this cluster with a gigabit bandwidth to run the passive receiver clients that are not measuring the performance.

First, we tested with multiple meetings each having 20 receivers. One client was publishing the video stream on a broker and 20 clients were receiving it. We distributed the clients of each meeting among brokers evenly. 5 clients joined each meeting through each broker. We also distributed the video transmitters of all meetings evenly among the brokers. There were equal numbers of transmitters publishing video streams to each broker. We collected the performance data from three meetings. The publishers of these three meetings were publishing their streams through the first broker. For each of these three meetings, we collected the results from 4 receivers, each one getting the stream from a different broker. Table 6 shows the average latency values of three meetings. Similarly Table 7, Table 8, and Table 9 show the results for jitters, percentages of lost packages and the percentages of late arriving packages, respectively.

Table 6 Average latencies for multiple video meetings each having 20 participants

Total meetings	Total users	Latencies from 4 brokers (ms)			
		B1	B2	B3	B4
24	480	3.30	4.24	5.32	5.87
48	960	2.98	5.04	6.90	8.20
72	1440	4.83	13.6	17.0	17.5
96	1920	5.76	25.5	52.7	47.3

Table 7 Average jitters for multiple video meetings each having 20 participants

Total meetings	Total users	Jitters from 4 brokers (ms)			
		B1	B2	B3	B4
24	480	1.39	1.44	1.65	1.72
48	960	1.84	2.53	2.82	3.00
72	1440	1.70	3.04	3.52	3.51
96	1920	1.69	3.70	5.28	5.52

Table 8 Average loss rates for multiple video meetings each having 20 participants

Total meetings	Total users	Loss rates from 4 brokers (%)			
		B1	B2	B3	B4
24	480	0.01	0.00	0.00	0.00
48	960	0.00	0.06	0.11	0.22
72	1440	0.02	1.20	1.60	1.63
96	1920	0.13	6.41	19.6	19.6

Table 9 Average late arrivals for multiple video meetings each having 20 participants

Total meetings	Total Users	Late arrivals from 4 brokers (%)			
		B1	B2	B3	B4
24	480	0.00	0.00	0.03	0.03
48	960	0.00	0.11	0.09	0.17
72	1440	0.00	0.01	0.03	0.04
96	1920	0.00	0.13	2.79	0.91

Since the publishers are publishing the streams through the first broker, the latency values for the first broker are the smallest. Latency values increase when the streams travel more hops along the way from the first broker to the last. The broker network provides excellent quality communication when there are less than 72 meetings. The latency values and jitters for all brokers are very small. There are minor package losses and late arriving packages. For 72 meetings, the latency values and jitters are still very small. There is also very few late arriving packages. However, there are a little more than %1.0 lost packages. When there are 96 meetings, significant amount of packages are lost. Therefore, the broker network can support up to 72 meetings or up to 1440 participants in total. This number is slightly smaller than the single video meeting case, in which the broker network was able to support up to 1600 participants.

When we compare the scalability of the broker network with the scalability of the single broker in multiple meeting tests at [9], the number of supported participants increased two times. The single broker supported 700 participants in 35 video meetings, each having 20 users. In this test, four brokers supported around 1440 participants in 72 meetings, each having 20 users. As we can see, the increase on the number of brokers did not result in a linear increase on the number of supported participants. There are two reasons for this. First one is the overhead of inter-broker stream delivery in distributed setting. Now, the brokers deliver streams not only to clients but also to other brokers. The second one is the smaller number of participants in each broker for each meeting. Each incoming package is delivered to only 5 users in the distributed setting, while it was delivered to 20 users in the single broker case. This test shows that the distribution of users in small size meetings among multiple brokers reduces the scalability of the broker network. When meeting sizes are small, it would be better to avoid distribution of users among brokers if possible. On the other hand, the distribution of users in large scale meetings among multiple brokers increases the scalability and the quality of the service as we pointed out in double queuing algorithm section.

Since the small number of participants joining meetings through each broker reduced the scalability and the quality of the service provided, we tested with a larger meeting size to observe the difference. This time, 10 participants joined each meeting through each broker. Therefore, the sizes of meetings were 40. All other aspects of the test were the same as the previous test. Below tables (Table 10, Table 11, Table 12, and Table 13) show the measured parameters.

In this case, the number of supported clients increased significantly. Now, 48 meetings with 1920 participants in total are supported with excellent quality, compared to 1440 participants in the previous test with meeting sizes of 20. In addition, the quality of the service provided by the broker network also increased considerably. The average latency and jitter values are much lower. The late arriving packages and losses are very small, too. The main reason for the better performance is the better utilization of the broker network. Now, there is less stream exchange among brokers and each incoming stream is delivered to more participants by every broker.

The scalability and the quality of service provided in this case are also much better than the single video meeting case on multiple brokers. Compared to the 1600 total users, now 1920 participants are supported. In addition, the latency and jitter values are much smaller. For 1600 participants, the latency values are less than 10ms, it was around 23ms for the single video meeting case for the last receiver. There is also a big difference in jitter values. While it is around 2ms for this case, it was more than 10ms for the single video meeting case. Therefore, this test demonstrates that similar to the single broker tests, it is possible to better utilize the distributed broker network by having multiple video meetings than by having a single video meeting as long as the distribution of clients among brokers and the meeting sizes are chosen properly.

Table 10 Average latencies for multiple video meeting tests each having 40 participants

Total meetings	Total users	Latencies from 4 brokers (ms)			
		B1	B2	B3	B4
40	1600	3.34	6.93	8.43	8.37
48	1920	3.93	8.46	14.6	10.6
60	2400	9.04	170.	89.9	25.8

Table 11 Average jitters for multiple video meeting tests each having 40 participants

Total meetings	Total Users	Jitters from 4 brokers (ms)			
		B1	B2	B3	B4
40	1600	1.15	2.20	1.99	2.10
48	1920	1.47	2.12	2.57	2.27
60	2400	2.42	4.62	4.81	4.60

Table 12 Average loss rates for multiple video meeting tests each having 40 participants.

Total meetings	Total Users	Loss rates from 4 brokers (%)			
		B1	B2	B3	B4
40	1600	0.00	0.00	0.00	0.00
48	1920	0.12	0.29	0.50	0.50
60	2400	0.16	1.30	2.51	2.82

Table 13 Average late arrivals for multiple video meeting tests each having 40 participants

Total meetings	Total users	Late arrivals from 4 brokers (%)			
		B1	B2	B3	B4
40	1600	0.0	0.00	0.00	0.00
48	1920	0.03	0.14	0.57	0.11
60	2400	0.00	53.0	36.4	0.69

We should also note the big difference among latency values from four brokers when there are 60 meetings. While the latency values from the first and the last brokers are relatively small, the latency values from the second and the third brokers are much higher. The main reason for this difference is the load on brokers. Since the broker network is organized as a chain, the middle two brokers in the chain delivers more streams among brokers. While the brokers on the edge delivers only the streams that are directly published on them by the clients to the next broker, the middle brokers both act as a bridge between the brokers on both sides and transmit the streams that are directly published on them to two other brokers on both sides. Therefore, this broker organization scheme puts more loads on the middle brokers. Other broker organization schemes can be explored such as ring or full mesh to avoid such disproportionate load.

6. WIDE AREA TESTS

In this section, we investigate the delivery of audio/video streams to geographically distant locations. We measure the quality of services provided in real life videoconferencing settings where clients can be scattered around the world. We had access to machines at three more locations, in addition to the two Linux clusters that we used for the previous distributed broker tests. Those two Linux clusters were in the same town (Bloomington, IN). The first one was in Community Grids Labs at IU (CGL) and the second one was in the department of Computer Science (CS) at IU. Both of these sites were connected to Internet2 with a gigabit bandwidth. The other three sites were in geographically distant locations: Syracuse University in Syracuse, NY, Florida State University (FSU) in Tallahassee, FL and Cardiff University in Cardiff, UK. All these three sites had 90 to 100Mbps download bandwidths.

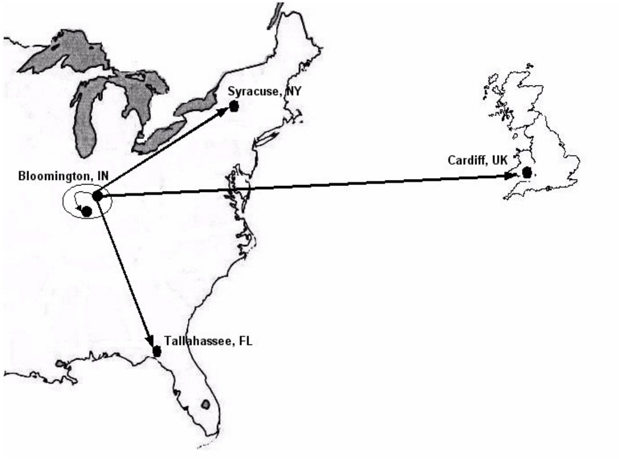

Figure 4 Wide are video delivery tests.

We tested the quality of service provided to remote participants in a video meeting with one broker. The broker was running at the CS site in Indiana and equal numbers of participants were running at the four other sites (Figure 4). A client running in the same site with the broker published the video stream on the broker. We measured the latencies, jitters and loss rates for the last clients at each site. We measured the clock differences between the transmitter and the receiver machines to calculate the latencies. There can be a few millisecond discrepancies on latency values because of the difficulties on determining the exact clock differences. Table 14 shows the latency values for the last users in four sites; Indiana (IN), Syracuse (NY), Florida State (FL) and Cardiff (UK). The last column shows the total amount of data transmitted to each site. Table 15 shows the jitters and Table 16 shows the loss rates for the same tests.

Table 14 Latency values for single video meetings with one broker in distributed setting

users per site	all users	Latencies of last participants				BW ps Mbps
		IN	NY	FL	UK	
4	16	1.8	13.28	13.63	56.36	1.2
50	200	10.84	23.94	24.42	65.55	15
100	400	21.36	36.25	36.78	76.15	30
150	600	33.04	49.8	47.6	86.98	45

Table 15 Jitter values for single video meetings with one broker in distributed setting

users per site	all users	Jitters of last users			
		IN	NY	FL	UK
4	16	0.6	0.77	2.42	1.55
50	200	4.87	4.45	4.39	4.52
100	400	9.95	8.98	8.92	9.43
150	600	14.8	13.4	12.9	14.8

Table 16 Loss rates for single video meetings with one broker in distributed setting

users per site	all users	Loss rates of last users			
		IN	NY	FL	UK
4	16	0	0.02	0.05	0
50	200	0	0	0.11	0
100	400	0	0.05	0.02	0
150	600	0	0.02	0.05	0

There are two major factors that contribute to the latency values shown on Table 14. These are the routing delays introduced by the broker and the transmission delays from the broker to the destination site. The latency values seen at Indiana is mainly due to the routing delays by the broker, since these two sites are very close to each other and connected with gigabit network. Therefore, we can estimate the transmission overheads for remaining three sites by subtracting the routing overhead (the latency for Indiana site) from their total latency. Similarly, these two factors are the main causes of the jitter, and we can estimate the jitter caused by the transmission by assuming that the jitter of Indiana site is mainly due to the routing at the broker.

The latency values at the first row show the minimum transmission times for packages to travel between the sites, since the number of participants is very low and the routing overhead is very small. The US sites have very low latency values, all are less than 14ms. That is excellent for video conferencing. Even latency values for UK clients are less than 60ms, which is excellent too.

When the number of clients at each site increases, both the routing overhead and the transmission latency increase. The routing overhead increases to 33ms for the last user with 150 participants at Indiana. However, the increase on the transmission overheads is much smaller for all three distant sites. It increases only around 5ms from 4 participants to 150. Similarly, the increase on the jitter is mainly due to the routing and the increase by the transmission is very small. Therefore, this test shows that 150 video streams can be transferred between these four sites with very small increase on the transmission delays and jitters. In addition, Table 16 shows that there are almost no package losses, even for the clients at UK when there are 150 users. These results indicate that very high number of video streams can be transferred between these four sites with excellent quality.

These tests demonstrate that by running a broker in a remote site, significant bandwidth savings can be achieved and the bandwidth limitations can be overcome to support more participants. If there was a broker running at each remote location in our video meeting test, only one copy of the video stream would have been transferred between the sites. This would reduce the used bandwidth and the load on the broker at Indiana significantly.

Another important observation about the results of Cardiff clients is the benefit of running brokers at geographically distant locations. Since the transmission times are significantly higher for geographically distant locations, it is very important to run brokers at those sites to minimize the transmission delays. For example, if the publisher was in UK for the single broker test, then the transmission overhead of the streams for clients at UK would have been increased by two fold. The video stream would have traveled through the Atlantic ocean two times to reach to the clients at UK. Therefore, it is critical to run brokers at geographically distant sites.

Maybe the most important result of the wide area tests is the fact that the networks that we tested provided very good quality communication for audio/video streams. When transferring even very high number of video streams, they provided excellent service for real-time videoconferencing applications. The lost rates were very small even for 150 video streams. Similarly, the amount of latency and jitter was very small. Even going through the Atlantic Ocean does not introduce a challenge. Therefore, the underlying network infrastructure is good enough to implement a distributed brokering system on top of it to deliver audio/video streams.

7. CONCLUSION

Our analysis of the performance of the broker network in distributed setting shows that there are many benefits of having multiple brokers. The first one is the support for higher number of users. In the case of large scale meetings, the number of supported users increases almost linearly by adding new brokers into the system. In our test setting, four brokers supported up to 1600 participants in a video meeting and this number can be increased easily by adding new brokers. In the case of multiple meetings, similar increase can be achieved as long as the distribution of users among brokers is managed properly. The second benefit of having multiple brokers is to provide better quality service with smaller latencies, jitters and loss rates to clients. Particularly the quality of stream delivery for large size meetings can be improved significantly by distributing clients among multiple brokers. In addition, running brokers in geographically distant locations can reduce the transit delays of packages considerably. The third benefit is the bandwidth savings. When there are multiple users on an organization, running a broker can reduce the used bandwidth significantly. In summary, the performance tests in this paper showed that GlobalMMCS videoconferencing system can be deployed in wide area networks to provide services to high number of users.

8. 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] G. Fox, W. Wu, A. Uyar, H. Bulut, S. 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] S. Pallickara and G. Fox. NaradaBrokering: A Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids. Proceedings of ACM/IFIP/USENIX International Middleware Conference. 2003. pp 41-61.

[8] A. Uyar, S. Pallickara, G. Fox, “Towards an Architecture for Audio/Video Conferencing in Distributed Brokering Systems”, The 2003 Int. Conference on Communications in Computing, June 23 - 26, Las Vegas, Nevada, USA.

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

[10] Sun Microsystems, Java Media Framework 2.1. 2001