3DTI Amphitheater

A Manageable 3DTI Environment with Hierarchical Stream Prioritization

Shannon Chen, Klara Nahrstedt, and Indranil Gupta
Department of Computer Science, University of Illinois at Urbana-Champaign
{cchen116, klara, indy}@illinois.edu

ABSTRACT

In this paper we present the 3DTI Amphitheater, a live broadcasting system for dissemination of 3DTI (3D Tele-immersive) content. The virtual environment constructed by the system mimics an amphitheater in the real world, where performers interact with each other in the central circular stage, and the audience is placed in virtual seats that surround the stage. Users of the Amphitheater can be geographically dispersed and the streams created by the performer sites are disseminated in a P2P network among the participants. To deal with the high bandwidth demand and strict latency bound of the service, we identify the hierarchical priority of streams in construction of the content dissemination forest. Result shows that the Amphitheater outperforms prior 3DTI systems by boosting the application QoS by a factor of 2.8 while sustaining the same hundred-scale audience group.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]; C.2.2 [**Computer Communication Networks**]

General Terms

Design, Algorithm, Optimization

Keywords

3D Tele-immersion, Broadcasting, Content Distribution

1. INTRODUCTION

Although being more elaborately advanced than conventional 2D video services, existing applications of 3D Tele-immersion (3DTI) are still restricted with a small number of immersive users. While we witness the thriving of live broadcasting video services such as PPLive [19], UStream [20], and Youtube [21], passive non-immersive audience is rarely included in the design of 3DTI systems. Hence, in this paper, we propose *the 3DTI Amphitheater*, a media-enriched multi-view live broadcasting system that renders a shared virtual space which mimics an amphitheater in the real world (Figure 1) that accommodates a hundred-scale user group.

Users in the 3DTI Amphitheater can be divided into two groups: performers and the audience. A user of an immersive site, or a *performer*, produces 3D streams by a 3D camera array

MMSys'14, March 19–21, 2014, Singapore, Singapore.

ACM 978-1-4503-2705-3/14/03…$15.00.
http://dx.doi.org/10.1145/2557642.2557654

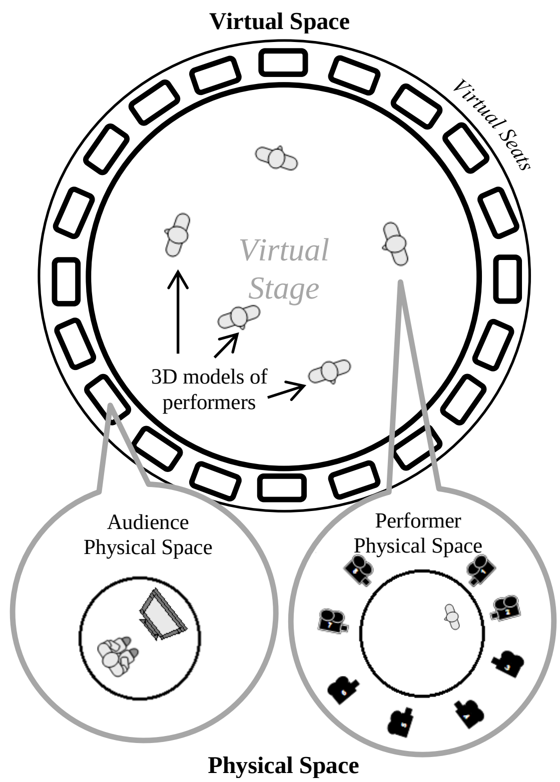


Figure. 1. An illustration of 3DTI Amphitheater.

surrounding her in her own physical user space. A 3D model of the performer will then be constructed from the streams and placed on the virtual stage to interact with other performers. Thus, the performer crew can be physically dispersed. On the other hand, users of non-immersive sites, or *the audience*, passively view the 3D streams created by the performers. They observe the interaction without any involvement. Every performer and every audience is a *viewer* in the amphitheater since they all need to view the virtual space from their different perspectives. In the virtual amphitheater, a performer's standing position on the virtual stage is synchronized with her position in her physical space so that she can move on the virtual stage freely as she moves in her own physical space. The virtual stage and the performer's physical space share the same size. The audience's position is fixed in pre-assigned *virtual seats* which surround the theater.

3DTI brings new challenges to the well-established IP-based live broadcasting framework. While IPTV service users only perceive one video stream created by one content provider (e.g., ESPN) at any time instance and rely on the service producer (session manager and cameramen) to choose the view, the 3DTI amphitheater has to support a much more elaborated immersive experience for its users. The first challenge is the multi-view feature. The multi-source (multi-performer), multi-content (multi-camera) dissemination transforms 3DTI broadcasting into a forest construction problem in the P2P network formed by all the participating sites (both performers and audiences are included). Multiple participants subscribe to multiple streams from multiple performers. This introduces massive bandwidth consumption for the dissemination. The 3DTI amphitheater tackles this problem by the design of *virtual seats*. The audience is placed in virtual seats that surround the central virtual stage and spread evenly, which makes the aggregated view of the audience covers 360 degree perspective of a performer. This implies two advantages in content dissemination. 1) The stream subscriptions of adjacent audiences have substantial overlap due to their similarity of views. Via content sharing in the P2P overlay, these highly overlapped subscriptions can be fulfilled together. 2) For each stream capturing a particular angle of a performer, there exists at least one audience who subscribes to it and thus can aid its distribution as a hub. This alleviates the pressure on the limited outbound bandwidth of the source performer site. These advantages help us outperforms the prior broadcasting framework [9] with a lower stream subscription rejection ratio under a series of hundred-scale user group simulations.

The second challenge is efficient delivery of the multi-view content. Although the camera array captures a performer with an omnidirectional perspective, a viewer simply does not require all the streams since she can only fix her view on one side of the performer at any given time. For instance, when the viewer is looking at the front of a performer, the streams capturing the back do not contribute to her view. This leads to the *differentiation of streams*: to a particular viewer, not all *cameras* are equally important. Prior works studying content delivery of 3DTI [5-9] stop at this point and adopt the contribution factor (CF) proposed by Yang *et al.* [7] to mark the importance of streams. The CF is defined as the inner product of the view direction and the shooting direction of a camera. Thus, a larger CF implies a more in-sync direction of the camera with the viewer's perspective, hence a more contributive stream which deserves a higher delivery priority. For the ease of discussion, we name this the *view-based priority* in the rest of the paper.

In this work, we further introduce the *differentiation of sites* to the dissemination problem. We argue that: to a particular viewer, not all *performers* are equally important. Intuitively, the importance of a particular performer depends on her *role* in the performance. The audience may be more interested in the vocalist of a rock band, the diva of an opera, or the quarterback of a football team. However, note that the importance of a performer may not be universal. It also depends on the *role* of the viewer herself. For example, the parents in the audience of a school play must be more interested in their own child on the stage. When the viewer herself is a performer, the importance of other performers may depend on the characteristic of the performance. For example, to mimic a realistic environment, the importance can be set inversely proportional to the distance between the viewer and the performer so that it can be dynamically adjusted according to the position of the viewer. As for gaming and sport scenarios, the importance of performers can be set equally, since every movement of each teammate/opponent has the potential to change the game. Thus, we define the *role-based priority* of a stream to capture its importance based on the semantic relation between its viewer (can be either an audience or a performer who requests the stream) and its performer (the producer of the stream).

Merging view-based and role-based priorities, we propose the *hierarchical stream prioritization*. We define *stream*, *view*, and *session*; each as an aggregation of its former. The definition of stream is straightforward, which is the real-time content produced by a 3D camera. A view is defined as a set of streams which are originated from the same performer; and a session is the set of performers in the amphitheater. Each viewer in the amphitheater has her own view-based priority and role-based priority stated in her *subscription request*. The construction of the content dissemination forest takes the hierarchical priorities into consideration, so that the important streams would not be discarded when the resource is critical.

We evaluate the performance of 3DTI Amphitheater by simulation with real-world network topology compiled by Netmap [10] and the configurations of real 3DTI testbed (TEEVE [11]). The evaluation consists of three parts. First, we examine the dissemination forest of 3DTI Amphitheater by comparing it with the forest constructed by 4D TeleCast [9], a prior framework of 3DTI broadcasting service. Results show that our framework can sustain the same number of users (100~1,000) with a decreased subscription rejection ratio (1,010 more stream subscriptions admitted while sustaining one thousand users). Second, we evaluate the effectiveness of our hierarchical stream prioritization by examining the application quality of service (AQoS) among immersive sites. The AQoS is defined as a weighted admission ratio of stream subscription request, where the 'weight' of a request is decided by the hierarchical priority of the requested stream (Section 6). We compare its performance with the random join admission algorithm adopted by Nahrstedt *et al.* in [8], which is a 3DTI dissemination scheme also based on the pub-sub model [18] as ours. Results show that the identification of view-based and role-based priorities helps boost up the AQoS by a factor of 2.8. Last, we investigate the role of audience in stream dissemination in the P2P network. To prove that the design of virtual seats actually contributes to substantial alleviation for content distribution from the performers, we simulate two amphitheaters with and without the audience. The result shows that the participant of the audience brings a 10% decrement in stream request rejection ratio and a higher AQoS.

In summary, the contributions of this paper include the following. 1) Proposition of a new live broadcast framework of 3DTI: the 3DTI Amphitheater, which makes use of the audience as relay sites to aid the dissemination of 3D streams. 2) Identification of role-based priority, which improves the utilization of limited bandwidth by giving higher priorities to the streams which are semantically more important to a viewer. 3) Proposition of the concept of hierarchical prioritization, which fine-grains the differentiation of streams and improves the resulting AQoS.

The remainder of this paper is organized as follows. In the first half of this paper, we set the stage by introducing our system after we review some related works in Section 2. Section 3 contains details of the system components, which classifies the users into three categories. Section 4 introduces the user model, which includes the amphitheater and the hierarchical prioritization. Section 5 introduces the stream delivery model, which contains the pub-sub model and the admission process of stream subscription. In the second half, we first describe the construction algorithm of our dissemination forest in Section 6, and then we evaluate our system in Section 7. Finally, in Section 8 we discuss a few possible improvements for future work before we conclude the paper in Section 9.

2. RELATED WORKS

IP-based Broadcasting. There are many IPTV frameworks such as [4, 14, 25] that aim for large scale dissemination of conventional 2D video streams. However, none of them consider multi-view dynamics with multi-source composition. Therefore, the challenges are different from ours. The 4D TeleCast framework which targets a large scale content multicasting of 3DTI is proposed in [9]. In order to accommodate the hundred-scale audience group, Arefin *et al.* propose to allocate the viewer sites into different classes (layers) with different delay service qualities. The higher the service class, the fewer hops there are between the viewer site and the source in the P2P network. The heavy burden of content dissemination not only comes from bandwidth and delay requirements, but also because they tried to support the randomness of the free-viewpoint characteristic. Viewers in the hundred-scale audience can all have their distinct views. This makes the effectiveness of content sharing very unstable in the dissemination network. Thus, the 4D TeleCast has to sacrifice part of the audience by giving them delayed content and requires an additional content distribution network to support the service.

Performer Differentiation. When an interactive multimedia service consists of multiple participants, the importance of content originated from different participants can have different importance to an observer. In [12], the concept is applied on an intelligent teleconference room. The room accommodates multiple participants, and is equipped with a fixed camera that captures the whole room. When a person starts to speak, she triggers the audio sensor near her seat. This notifies the video encoder to allocate higher resolution to the speaker's position in the scene.

3DTI Multicast Routing. The differentiation of 3DTI streams introduces extra complexity to the construction of the content sharing overlay. Because every link in the overlay network is transmitting different parts of the stream bundle [6] (a combination of streams containing different sensing data that are highly correlated), the topology of the network becomes a crucial factor that decides the efficiency of content dissemination. In [5], a mesh topology is used to deliver the content from each producer site to one another. The number of participating site therefore becomes very restricted (fewer than four). Later, [2] proposed to alleviate the workload via a randomized admission under a pub-sub model. Yet, the algorithm failed to consider the role of the users and focused only on differentiation at stream level.

3. SYSTEM MODEL

In our system model, multiple 3DTI sites collaborate together to produce and distribute the 3D visual content. Among all the participating sites, only a subset of the sites is producing the 3DTI video streams. Users within these sites have their 3D models projected into the virtual space where they interact with each other. In the rest of the sites, the users only passively observe the activity from the view they choose and do not have their 3D models built by the system. Together, these sites form a P2P overlay network that delivers the content streams. An example of a use case is illustrated in Figure 2a. Inside the virtual space depicted in the figure, only three performers have their 3D models created (P1, P2, and P3). These performers actively interact with each other through their model on the virtual stage. Outside the space, there are two other audiences (A1 and A2) passively observing the interaction from the view point they get assigned. Their sites do not produce any 3D visual contents but only passively receive them from other sites. Based on the different roles of a participating site in a 3DTI session, we classify them into three types: performer sites, audience sites, and the session management site. In the following, we introduce the system requirements and the assignments of each type.

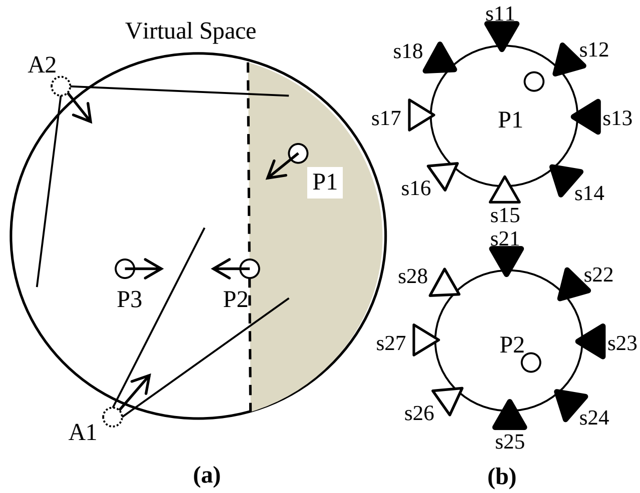


Fig. 2. An example of (a) virtual space and (b) physical space.

3.1 Performer Sites

A performer site is a producer of the 3DTI content during a session and also a consumer of the content for other performers. As mentioned, users in the performer sites are the performers that collaborate with each other. Thus, the hardware requirements of a performer site include a 3D camera array, a head mounted display, and a gateway (GW) machine (Figure 4a).

The 3D camera array consists of a group of 3D cameras that surround the user space (Figure 2b). Each camera captures a facet of the object in the physical user space of the site. In real-time, the streams produced by the cameras are delivered to the gateway machine located in each participating site to render the 3D model of the performer. The gateway machines in the performer sites receive streams produced by all the performer sites and render out a consistent virtual space. According to the standing position and the view direction of performer in the physical space, the head mounted display connected to the gateway shows the relative view of the virtual space. The position of the user is calculated based on the 3D information captured by the camera array while the view direction is captured by the triaxial accelerometer attached to the head mounted display.

Due to the interactive characteristic, the end-to-end delay of content delivery among performer sites is tightly bounded (less than 100ms). With the restricted networking resource (40~150 Mbps bandwidth for typical 3DTI sites) and the media-rich visual streams (5~10 Mbps per 3D video stream), the dissemination of content among performer sites is not trivial (Section 6).

3.2 Audience Sites

The audience sites are observers during a 3DTI session. Users in audience sites passively view the performance without any involvement. Thus, the 3D camera array becomes optional within an audience site. The basic hardware requirement for the audience site only includes the gateway machine and a display. The gateway machines in audience sites collect the streams produced by the performer sites and render the 3D virtual stage space. The virtual stage space is shown by the display in the audience site, which can be either a conventional display or a head mounted one.

For a passive audience, the end-to-end delay bound of content delivery can be loosened. According to [13], delivery delay of a non-interactive multimedia broadcasting service as PPLive varies from twenty to 120 seconds based on the popularity of the channel. However, in 3DTI, we have to provide the 'view change' feature to our users, even if they only passively receive. Although viewers in the audience have their positions fixed in their virtual seats, they still have the ability to change their view directions to focus on a particular part of the stage (depicted by the triangular areas coming from A1 and A2 in Figure 2a). Hence, the reaction time of view change bounds the end-to-end delay under five seconds to avoid degradation of the user experience.

3.3 Session Management Site

In each 3DTI session, one site is chosen by ring-based election [3] to become the session management site. The job of the session management is to examine the *subscription requests* (Section 5) sent from all the other participating sites. The subscription request contains information to determine 1) which streams the subscriber requires and 2) what the hierarchical priorities (Section 4.2) of the subscriber are. With these requests, the session manager can construct the dissemination network using our forest construction algorithm (Section 6). In reply, the session manager will tell the subscriber from whom should it receive the required streams. More details on this topic will be introduced in Section 5 and 6.

4. USER MODEL

In this section we describe our user model in two parts. First, we introduce the characteristics of the Amphitheater and how its structure effects the subscription and dissemination of streams. Second, we introduce the hierarchical stream prioritization, which combines the view-based and the role-based priorities of a viewer.

4.1 The Virtual Amphitheater

The virtual space constructed mimics an amphitheater, where the performers are interacting on the central virtual stage and the audiences are assigned with their own virtual seats which surround the stage and disperse evenly. Thus, the perspectives of a performer and an audience are different. While a user in a performer site may not be able to see the whole performer crew due to her standing position and view direction, the users in the audience site can always choose to capture the whole view of the virtual stage or to focus on part of the stage. As illustrated in Figure 2a, where the arrows indicate the view directions, performer P2 can only see P3 but not P1 since P1 is standing in her blind side (the grey area). As for A1 and A2 in the audience, A2 choose to see the whole stage while A1 focuses on a particular part as if she sees through a pair of opera glasses.

In addition to the visibility of the performers, another factor that affects the user view is the relative position of a viewer-performer pair. At any given time, a viewer can only see a facet of a performer. Thus, the viewer site does not need all the streams originated from that performer site since half of the streams capture the opposite facet and do not contribute to the view. An example is provided in Figure 2, where the streams are denoted by the cameras in the performer sites (Figure 2b). According to the relative positions, performer P3 only requires stream s15, s16, s17 from P1 and stream s26, s27, s28 from P2.

The virtual seats fix the position of each audience. Although this design restricts the audiences from moving their viewpoints freely inside the whole virtual space (e.g., around the stage or even on stage), it complies with the common sense in a real theater, where seats are pre-assigned and fixed during the performance. Furthermore, the design brings two advantages to the delivery of streams. First, it enhances the effectiveness of content sharing. For two audiences in adjacent seats, their views are very likely to overlap with each other by a fair portion. This implies that in stream delivery, the same stream is more likely to be subscribed by multiple sites, which makes the sharing of the content being able to save more bandwidth in the P2P network. For example, in Figure 2, while audience A2 subscribes to stream s27, *s28*, *s21*, an audience A2' sitting on her left possibly subscribes to *s28*, *s21*, s22, hence a good portion of required streams can be shared between them.

Second, the surrounded arrangement of seats helps the distribution of streams from their source sites. Since the stage is surrounded by the audience, each and every side of the performer's body must be looked at by some audiences at any given time (note that occlusion is overseen in this statement). This implies that every stream (each capturing different sides of the performer) is likely to be subscribed by one or more audience sites at any time. In the dissemination network, these audience sites act like hubs that help the performer sites to distribute their streams to other performers. Often times the outbound bandwidth of a performer site is not enough to distribute its streams to all the other performers when the performer crew grows big. In that case, audience sites can help as hubs to relay those contents. The surrounded seat design raises the possibility of the existence of such hub audiences.

4.2 Hierarchical Stream Prioritization

The view-based priority and the role-based priority are both addressed in the hierarchical prioritization of streams. We introduce the three logical objects that we define in the hierarchy as follow (using Figure 2 as an example):

- **Stream.** A 3D video stream created by a camera. This is the basic content unit in the dissemination network. We denote a stream by *s* with a postfix number for identification, e.g., s22.
- **View.** The set of streams that are created in the same performer site. We denote a view by *v* with a postfix number to identify the site, e.g., v2 = {s21, s22, s23, s24, s25, s26, s27, s28} is the view of performer P2 in Figure 2.
- **Session.** The set of views in the amphitheater. We denote a session by *x*, e.g., x = {v1, v2, v3}.

As we discussed previously, there are two factors that affect the importance of a stream to a particular viewer. First, the view-based priority reflects the importance of a camera per site. If its shooting angle complies with the view angle of the viewer, then the stream it produced becomes more important to the viewer. Second, the role-based priority reflects the importance of performer per session. If a stream captures a performer in whom the viewer is more interested, then the stream is more important to the viewer.

To address these factors, each viewer would provide information to determine her own *hierarchical priority* in her subscription request (Section 5). The hierarchical priority is represented as a sequence of numbers assigned to each element in a view or a session. For example, to address the view-based priority, a viewer may set her hierarchical priority as:

HP(v2) = { 0, 0, 0, 0, 3, 4, 3, 0}
for v2 = {s21, s22, s23, s24, s25, s26, s27, s28}

This hierarchical priority (HP) states that, for this viewer, stream s26 is the most important stream among all streams in the set v2; s25 and s27 come second; and the viewer does not care about s21~s24 and s28. A larger number indicates higher importance of the stream. The method we apply to determine the numbers for view-based priority is modified from the contribution factor (CF) proposed by Yang *et al.* [7]:

$$CF = \overrightarrow{O_\iota} \cdot \overrightarrow{O_u}$$

where $\overrightarrow{O_\iota}$ is the shooting direction of a camera, $\overrightarrow{O_u}$ is the view direction, and CF is defined as their inner dot product. Our priority numbers are calibrated from the CF value by 1) treating non-positive CF as zero and 2) normalizing the numbers so that their sum becomes ten. The calibration is only for the ease of computation in the construction of the dissemination forest. (Section 6).

To address the role-based priority, the hierarchical priority is defined similarly, as a mapping from the views in a session to numbers, e.g.,

HP(x) = {5, 3, 2}
for x = {v1, v2, v3}

This states that, for the viewer who assigned this HP, the view that captures performer #1 (v1) is more important than v2 and v3; and v3 capture the least important performer. Again, the numbers are non-negative and they are normalized so that their sum is ten for the ease of computation. The determination of the numbers depends on the role of the viewer and the scenario. As we previously introduced in Section 1, there are many ways to determine the role-based priority. Here we provide three examples for three different roles of a viewer.

Viewer Role 1: Parent in the audience of a school play.

HP(x) = {1, 8, 1}
for x = {v1, v2, v3}

The priority numbers are subjectively assigned by the viewer (parent). Since v2 is capturing the image of the viewer's child on stage, she assigns the highest priority to it and assigns the others kids with one.

Viewer Role 2: Player in a table tennis dual match.

HP(x) = {3.33, 3.33, 3.33}
for x = {v1, v2, v3}

In this case, the viewer is a performer. The priority numbers are uniform. For v1 being the viewer's teammate and v2, v3 being the opponents, they have equal possibilities to change the game. Thus the role-based priority numbers are the same for all three views in the session.

Viewer Role 3: Participant of a cocktail party.

HP(x) = {1/the distance from performer #1,
1/the distance from performer #2,
1/the distance from performer #3,}

for x = {v1, v2, v3}

In this scenario, all the participants (performers) interact with each other in a ball room (stage) that is monitored by the security (audience). The priority numbers are objectively determined based on the distance between two participants in the virtual environment. This way, the view of a performer who stands closer to the viewer will be given higher priority than the ones that are far away. Higher priority implies higher admission rate of streams in the dissemination forest construction (Section 6) and thus a higher quality of the performer's image. This complies with the sense in the real world, where it is hard to see clear of a further object but easy to see an object clearly when it is close to you.

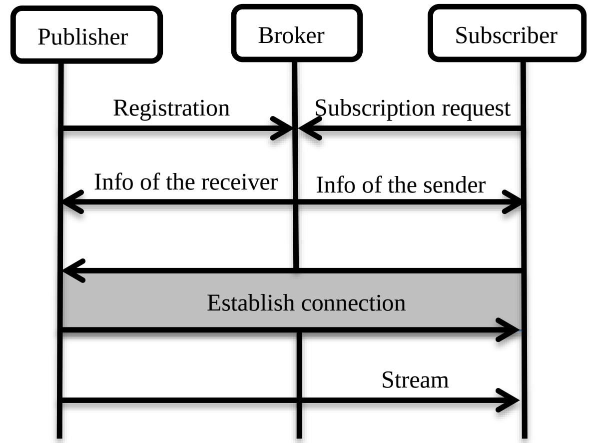


Fig. 3. Protocol in the pub-sub model.

The three examples show three possible ways to determine the role-based priority: 1) user-defined, 2) uniform, and 3) objective. Note that these are not the only methods but rather intuitive examples. Depending on the characteristic of scenarios and roles, we expect more creative determinations so that our system can become TEEVE (Tele-immersive Environment for EVErybody).

5. STREAM DELIVERY MODEL

Our Amphitheater system adopts the pub-sub (publish-subscribe) model [18] as its content dissemination paradigm. The model has three core components: publisher, subscriber, and broker. In the Amphitheater, the publishers are the performer sites; the subscribers are all the viewers (including both performer and audience sites); and the broker is the session management site. We introduce the message exchange among the three components as follows. An illustration of the whole process is shown in Figure 3.

In the beginning of a session, the publishers register their cameras to the broker. The *registration* states the number of cameras that the performer owns and their shooting angles. The subscribers, on the other hand, submit their *subscription requests* to the broker, which contains the type (performer or audience), the state (position and view direction), and the interests (user-defined role-based priority) of itself. Whenever there is an update on the subscriber's information (e.g., change of position), it renews the subscription request and sends it to the session manager again. Registrations and subscriptions also contain the geographic location and the maximum inbound/outbound bandwidth of the sender. This information is used to estimate the propagation delay between sites and will be used in the construction of dissemination forest (Section 6).

After the broker receives all the registrations from the publishers and all the subscription requests from the subscribers, it starts to translate them into *stream requests*. Since the broker knows 1) the positions and view directions of every viewer, and 2) the positions of each performer and the shooting angles of their streams, it can deduct which streams would be needed by a particular subscriber in order to construct her view. For example, in Figure 2, after the broker gets the positions of all participants (including all audiences and all performers), it will know that stream s26 of performer P2 is needed by audience A1.

Furthermore, the broker would know how important a stream is to the particular viewer by considering its view-based and role-based priorities. The determination of view-based priorities is the same as the transformation from a subscription request to stream requests we previously introduced. Since the broker knows the position and view information of each viewer-performer pair, it can infer the view-based priority for each stream request. In Figure 2, for audience A1 who has a 45 degree view direction (we set 0 degree to be pointing to the right), the CF of s26 is $\cos 0° = 1$ because s26 also shoots with a 45 degree direction. CF of s25 for A2 is $\cos 45° = 0.707$ because there is a 45 degree difference between A2's view direction and s25's shooting direction.

The determination of role-based priority, on the other hand, depends on how it is defined as we discussed earlier in Section 4.2. Here, we use the three viewer roles in the discussion as examples. For the first two determination methods: user-defined and uniform role-based priorities, the broker needs no extra computation to get the priority numbers since they are already specified in the subscription request (the interest field). As for the last objective determination method which is based on distance, the computation requires the positions of the viewer and the performers in the virtual space. This information is already stated in the subscription requests (the state field). Thus, the role-based priorities of all the subscribers are also known by the broker.

After the stream requests and the hierarchical priorities are determined, the final job of the broker is to construct the content dissemination forest and then notifies the subscribers from whom should they receive the streams. We leave the construction details for the next section. Note that a subscriber does not necessarily receive streams from a publisher as depicted in Figure 3. Instead, it may receive the stream from another subscriber who needs the same content and helps to forward the stream.

6. DISSEMINATION FOREST

The content dissemination forest decides the efficiency of stream delivery. Under the pressure of massive audience population, bandwidth-consuming streams, and tight delay bound, an efficient dissemination network would help preserving the service quality under the limited resource.

Now we continue the job of the broker (session manager) after it acquires the stream requests and the hierarchical priorities of the viewers. The objective is to construct a set of directed trees (a forest) in the P2P network among the participating sites (every performer and every audience site is included). Each tree connects all the subscribers who require the same stream with the publisher (performer) of that stream as the root. An example is provided in Figure 4b. We simplify the situation by assuming each performer only produces one stream. Under resource limitations, often times not all the stream requests can be admitted. Thus, construction of an efficient forest which preserves a low request rejection ratio is crucial.

Another important metric that reflects the efficiency of a forest is its resulting application quality of service (AQoS). AQoS is a weighted version of the admission ratio of stream requests (formal definition will be provided in problem formulation). From hierarchical priorities, we know the importance of a particular stream to its subscriber. The more important the stream is, the higher its 'weight' will be. Thus, the AQoS metric essentially shows whether the forest construction algorithm can identify the

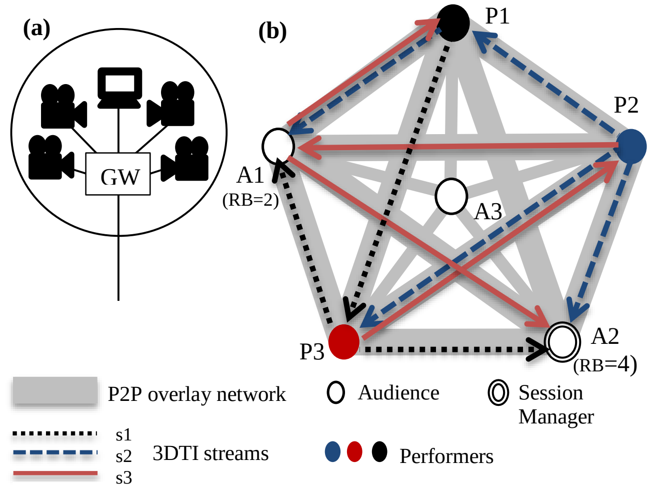


Fig. 4. Content dissemination forest.

importance of the content and can choose the less important ones to discard when there is not enough resource. In the following, we formulate the forest construction problem and introduce our construction algorithm.

6.1 Problem Formulation

The problem contains two constrains and two optimization goals.

Bandwidth Constrain. For each participating 3DTI site U, there is a limit on the inbound and outbound bandwidth of its local gateway machine, denoted as I_U and O_U. These limits can be measured by various probing tools (e.g., Pathload [15]). The total bandwidth consumed by streams received by the site must not exceed the limit I_U; and the total bandwidth consumed by streams going out from the site (including those produced by the site itself and those relayed by the site) must not exceed the limit O_U.

Latency Constrain. To preserve the real-time property of 3DTI service among performer sites, an end-to-end latency bound DI is placed on the content delivery to a performer site. As for audience sites, since a delay of content delivery in a reasonable range (< 5s) does not degrade the service quality, the latency bound for audience sites, denoted as DP, is typically larger then DI. The P2P overlay network among participating sites is a complete graph with a cost on each of its edges. The cost denotes the time delay for a stream to travel from one end to the other. These costs are estimated by the geographic locations of the sites by the empirical mapping provided in [16]. In forest construction, the total cost of a delivery route of a stream must not exceed the latency bound assigned (DI or DP). Note that the delay bound also constrains skew and synchronization problem. The inter-arrival time between two streams must be less than the delay bound if they both comply with the delay constrain. The streams will be buffered until the last one arrives and then they will be played out together. Since the inter-arrival time is constrained, the maximum buffer size is constrained.

Minimizing the Stream Request Rejection Ratio. Any stream request that violates either one of the mentioned constrains will be rejected by the session management site at forest construction. Since every request rejection implies potential degradation of the final service quality, our first goal of the construction is to minimize the number of rejected stream requests. Thus, we define the *Request Rejection Ratio* of streams as the number of rejected stream requests (*Nr*) over the total number of stream requests (*N*): *Nr/N*.

Maximizing the Application Quality of Service (AQoS). With the view-based and role-based priorities of a subscriber defined in Section 4.2, we define the *hierarchical priority (hp)* of a stream request as the product of 1) the priority number of the stream in view-based priority, and 2) the priority number of the view that contains the stream in role-based priority. For example, if for a particular viewer:

$$HP(v2) = \{0,\ 0,\ 0,\ 0,\ 3,\ 4,\ 3,\ 0\}$$
$$\text{for } v2 = \{s21, s22, s23, s24, s25, s26, s27, s28\}$$

and

$$HP(x) = \{1,\ 8,\ 1\}$$
$$\text{for } x = \{v1, v2, v3\}$$

then the *hp* of stream s25 is 3×8=24 ("HP(v2) of s25" × "HP(x) of v2" = *hp*). Since the priority numbers (elements of HP(·)) range from zero to ten, the resulting *hp* ranges from zero to 100 ($hp \in [0,100]$). Note that if either of the priority numbers is zero, it means 1) this stream does not contribute to the view at all, or 2) the subscriber has no interest in the performer captured by this stream. In either case, the resulting zero *hp* signifies the session manager to ignore this stream request. We define the AQoS as:

$$\text{AQoS} \equiv \frac{\text{sum of } hp \text{ of the admitted requests in the session}}{\text{sum of } hp \text{ of all requests in the session}}$$

The resulting value is between zero and one, with one being the highest quality (the subscribers get all the streams they request), and zero being the least (none of the requested streams is admitted).

6.2 Forest Construction Algorithm

In [17], Wang *et al*. proved that when a multicast routing problem is bound with two or more orthogonal constrains (in our case: bandwidth and latency), it becomes a NP-complete problem. Thus, we propose a heuristic solution based on the *hp* of the stream requests. The main complication of our forest construction problem is two-folded. 1) Which request should be examined by the session manager first? 2) Whom should the subscriber receive the stream from when there are multiple holders of the requested content?

Order of Request Examination. The order of stream request examination decides the possibility for a specific request to be admitted. Intuitively, the first request being examined should always be admitted since the bandwidth of the overlay network has not been occupied by any other delivery. The later a request is examined indicates the higher the chance should it be rejected since the links could be occupied by preceding requests. Thus, we order the stream requests of all the subscribers by the *hp* of the requested stream from high to low. We examine the higher *hp* (high importance) stream request first to grant it a higher probability to be admitted.

Selection of Sender Site. Since a stream is shared among the subscribers in the same tree via relay, there can be more than one site caching the same stream. For example, in Figure 4b, if A3 has a request of stream s1, the potential sender sites are P1, P3, A1, and A2 since they all hold s1. Our selection strategy can be broken down to three heuristics: similarity, residual bandwidth, and distance, ordered by the sequence of application.

Heuristic 1: Similarity. According to the type (performer or audience) of the subscriber, it will be assigned to the sender of the same type if possible. This helps the performer sites receive streams from another performer instead of the audiences. The end-to-end latency among performers can be reduced because the performer sites will be closer to the root (the source performer) in dissemination trees with this heuristic. Since the audience sites would also receive streams from their own kind over the performer sites under this heuristic, the outbound bandwidth of performer sites can be reserved for other performers.

Heuristic 2: Residual Bandwidth. The subscriber will be assigned a sender with the most residual bandwidth. For an audience site, the residual bandwidth equals to its maximum outbound bandwidth O_U subtracted by the bandwidth consumed to relay streams. As for performer sites, the session manager has to make sure that all the requested streams can be sent out from its performer to at least one other site, or else no subscriber can receive this stream. Thus, the residual bandwidth is further deducted by a reserved bandwidth. The reserved bandwidth of a performer site is computed as the sum of bitrate of streams that fits all the following three conditions. 1) The stream is produced by the performer itself. 2) The stream is requested by at least one subscriber. 3) The stream has not been sent to any other site yet. This strategy is originally proposed in [2], which guarantees that a requested stream can be disseminated before the outbound bandwidth of its producer is saturated.

Heuristic 3: Distance. On tie-breaking of the previous heuristic, the sender is set to be the one who is closer to the root (the performer of the stream) in the dissemination tree. This shortens the end-to-end delay.

We now use Figure 4b as an example to demonstrate the selection of sender site. We simplify the problem in this example by assuming the inbound and outbound bandwidths of all sites are able to sustain no more than four streams (assuming homogeneity of stream bitrates). The session involves six sites in total, which includes three performer sites (P1, P2, P3) and three audience sites (A1, A2, A3). The RB in the figure indicates residual bandwidth (in number of streams it can sustain) of the site. Following the order of request examination, let the next three stream requests to be examined by the session manager to be "A3 requesting s1", "A3 requesting s2", and then "A3 requesting s3". Thus, by our selection strategy, A2 will be the assigned sender for s1 because it is the audience site (by Heuristic 1) with the most residual bandwidth (by Heuristic 2). After this assignment, the RB of A3 becomes three. Next, A2 will also be the sender for s2 for the same reason and RB of A3 becomes two. Finally, the sender of s3 will be A1 because now the two audience sites (by Heuristic 1) have the same residual bandwidth (by Heuristic 2) but A1 is closer to the root of the tree that disseminates s3 (by Heuristic 3).

7. EVALUATION

The evaluation of our system is three-folds. First, we evaluate the overall performance of the Amphitheater with hundred-scale audience group. Second, we focus on the service quality of the performer sites. Last, we verify the effectiveness of virtual seat design on improving the efficiency of stream dissemination. In the following sections, we first introduce the settings of our experiment testbed before we continue to the three evaluations and their result analysis.

7.1 Simulation Settings

Network Settings. We adopt real-world topology from Netmap [10] as the testbed of our simulation. Among the 1,092 real hosts distributed around the world in the Netmap database, we randomly picked 3 to 1,000 hosts to be our participating 3DTI sites. The host-to-host delay is estimated based on the geographic distance between them [16]. We set the maximum inbound (I_U) and outbound bandwidths (O_U) of a site to be random values in the range of 40~150 Mbps based on our observation of the TEEVE system.

Site Settings. Each participating performer site is equipped with a camera array with eight 3D cameras shooting from octagonal positions around the user space. According to our observation of the TEEVE system, each camera produces a 3D video stream with a 5~10 Mbps bitrate. Hence, in the simulation, the bandwidth consumption of a stream is set to be a random number in that range.

Latency Settings. The latency bound is set to be 100ms (*DI*) for performer sites and 5s (*DP*) for audience sites. Contents should be delivered with an end-to-end delay exceeding the bound will not be admitted by the session manager.

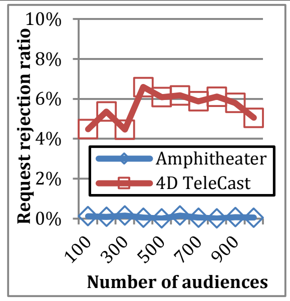
7.2 Broadcast with Hundred-Scale Audience

In this part of the evaluation, we evaluate the overall performance of the Amphitheater as a hundred-scale broadcast system of 3DTI. We investigate the effect of the size of performer crew and the size of the audience in the two parts of this experiment. The simulation results are compared with 4D TeleCast [9], a prior 3DTI dissemination system that also targets on large scale broadcasting.

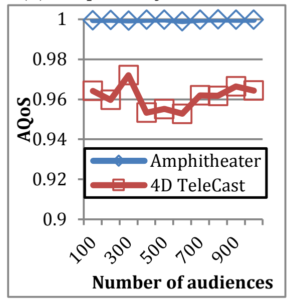
Performer Settings. On the circular stage of the Amphitheater, we randomly assign the standing positions and the view directions of the performers. The role-based priorities are set according to the distance between the viewer and the targeted performer (Viewer Role example 3 in Section 4.2).

Audience Settings. The audience is put in virtual seats, which surround the stage and are evenly dispersed. We set the views of the audiences to be pointing towards the center of the stage and the field of view covers the whole stage (the same as audience A2, depicted in Figure 2a.) The role-based priorities are set according to the 'popularity' of the targeted performer. In the performer crew, we set 25% of the performers as being 'popular', 50% of them being 'average', and 25% of them being 'subordinate'. The role-priorities assigned by the audience are random variables. However, we set the average value of the assigned role-priorities to be the highest for popular performers; and the lowest for subordinate performers. This setting is a more sophisticated version of the Viewer Role example 1 in Section 4.2.

Simulation Results. In the first part, we fix the number of performer sites at five and gradually increase the total number of audience sites from 100 to 1,000. The results are plotted in Figure 5 with the performance of the Amphitheater being the blue-diamond curve and 4D TeleCast being the red-square curve. We can see that the number of the audience sites does not affect the rejection ratio significantly. Intuitively, more sites participating in the session implies taller stream dissemination trees (since the outbound bandwidth is limited) and hence longer latency. However, since non-interactive audience has a much higher tolerance towards delay (DP = 5s), the growth of the audience group does not increase the stream request rejection ratio in both of the frameworks. The rejection ratios are lower than 10% for

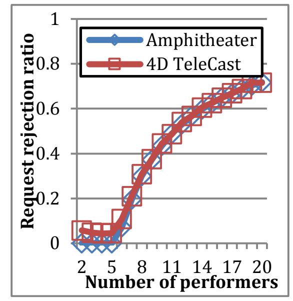


(a) Request rejection ratio

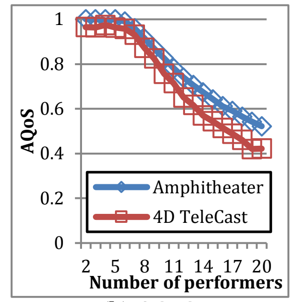


(b) AQoS

Fig. 5. Broadcasting with fixed performer sites.



(a) Request rejection ratio



(b) AQoS

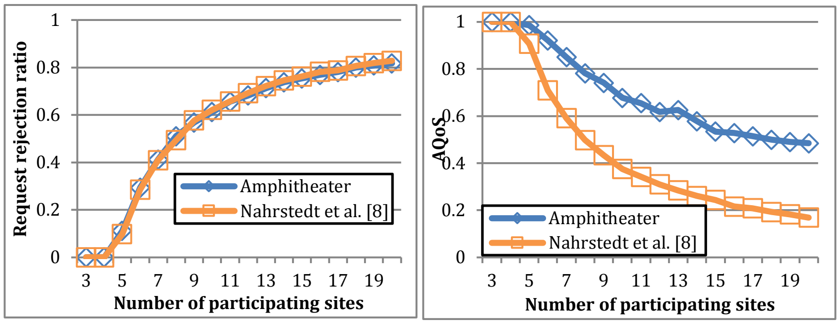
Fig. 6. Broadcasting with fixed audience sites.

both cases and the AQoS are higher than 0.9 due to the high request admission rate.

Comparing the performance of the two frameworks from Figure 5, we can see that even with abundant resource, 4D TeleCast still has a slightly worse performance on both metrics. The phenomenon is caused by the 'number of accepted stream constraint' in the design of 4D TeleCast. Since role-based priority is not identified in 4D TeleCast, its scheduling algorithm assumes equal importance of all performer sites and will 'drop a viewer' (by rejecting all of its stream requests) when it cannot receive at least one stream from each performer. On the other hand, since our examination order of stream requests inherits the role-based priority, if a viewer does not receive any stream from a particular performer, it is because she has marked that performer as unimportant with a low role-based priority number.

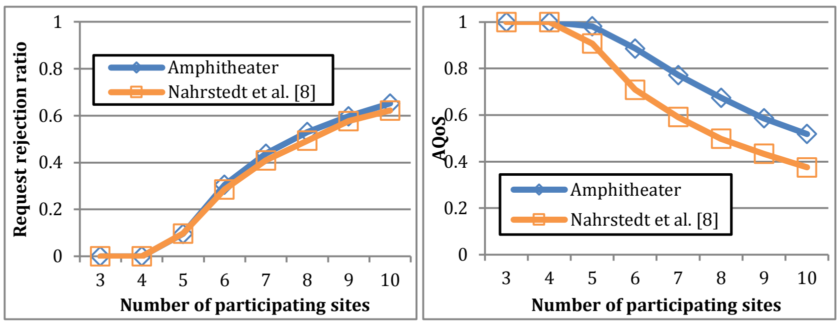
The all-or-nothing (have to receive stream from each and every performer or else the viewer is dropped) design of 4D TeleCast oversees the fine-grained hierarchical priority of streams. This contributes to an inferior performance when the user body grows. When the size of audience reaches one thousand, the Amphitheater sustains 1,010 more stream requests than 4D TeleCast with higher AQoS.

In the second part, we conduct another set of simulation with a fix number of audience (500 sites) and performer crews with different sizes (two to twenty). The results are shown in Figure 6. We can see from the figure that our forest construction algorithm is able to identify the important requests and to assign resource accordingly. With no significant difference (< 6%) between the two frameworks on request rejection ratios, our algorithm is able to reach a higher AQoS. The improvement from adopting our algorithm grows with the increasing pressure of resource limitation. When the number of performers reaches twenty, the total bitrate of streams created by the whole performer crew is 1,600 Mbps, and our algorithm achieves 24% more AQoS than 4D TeleCast.



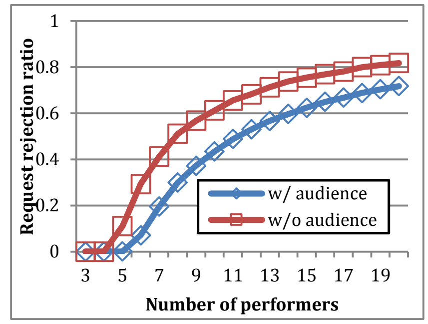
(a) Request rejection ratio **(b)** AQoS

Fig. 7. Simulation results of virtual play.

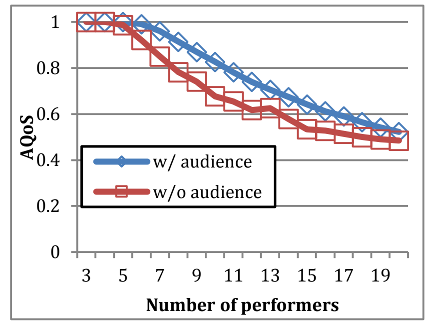


(a) Request rejection ratio **(b)** AQoS

Fig. 8. Simulation results of sport arena.



(a) Request rejection ratio



(b) AQoS

Fig. 9. Performance with and without audiences.

7.3 Service Quality of Performers

In this part of the evaluation, we focus on the performer sites and evaluate their service qualities. The quality of a performance depends largely on the quality of interaction among the performers on stage. We compare our result with the forest construction algorithm used in the framework proposed in [8] by Nahrstedt *et al.*, which assumes every participating site is immersive and hence every user (in a small user group) is a performer in their scenario. Thus, in the simulation, we accommodate the Amphitheater to this scenario by setting the auditorium to be empty. There are only interactive users in the session.

Performer Settings. The evaluation contains two parts. In the first part, we set the stage at a virtual play, which contains fewer than twenty performers, and the role-based priority of a viewer is set according to the distance (Viewer Role example 3 in Section 4.2). In the second part, we set the stage at a sport arena. Where fewer than ten performers (athletes) are in the arena and the role-based priority of a viewer is set to be uniform (Viewer Role example 2 in Section 4.2). In both scenarios, the performers are placed in random positions on the stage and each of them has a randomly set view direction.

Simulation Result. Since all participating sites are performer sites, intensive stream exchanges and hence massive bandwidth consumption in the overlay network are well expected. The results of virtual play and sport arena are presented in Figure 7 and 8, respectively.

In Figure 7a and 8a, the request rejection ratios of the two algorithms are plotted against the number of participating sites with the results of the Amphitheater being the blue-diamond curve and [8] being the orange-square curve. First, we can see the ratios increase along with the number of the participating sites in both cases. With the constant networking resource, the increasing number of sites is introducing more stream requests that cannot be admitted due to the lack of bandwidth or to the violation of latency constraint. From the figure we can see that for cases with fewer than five participating sites, the network can sustain all of the requests addressed by the users. After that, the rejection ratio rises gradually. The rejection ratio is nearly 80% when the number of sites reaches twenty in the virtual play; and 60% when it reaches ten in the sport arena.

Second, comparing the performance of the two systems, we can see the resulting rejection ratios are very similar. In [8], the authors adopt the random join algorithm to construct their content dissemination forest. The basic idea is to examine the stream requests in random order to avoid biased resource allocation to any content producer or subscriber. In our algorithm, the hierarchical priority provides enough randomness to the order of stream request examination. The standing position of a performer in the user space (which affects the view-based priority) and the different roles of the performers (which affect the role-based priority) are independent of the underlying resource demand. Hence, the *hp* of a stream request, defined as the product of view-based priority number and role-based priority number, provides sufficient randomness for the low-level resource allocation. In other words, the *hp* value is no different from a random value in forest construction. Hence, the two algorithms achieve similar request rejection ratios in all cases (difference < 4%).

Last, we look at the AQoS of the two algorithms in Figure 7b and 8b. Since AQoS is actually a weighted version of the admission ratio of stream requests, the curves of AQoS have opposite shapes of rejection ratio. As the number of sites increases, AQoS drops intuitively. Comparing the two algorithms, we can see that our algorithm outperforms [8] by a factor of 2.8 when the number of sites reaches twenty in the virtual play, and 1.4 when the number of sites reaches ten in the sport arena. This shows that, although the two systems has rejected the same amount of stream requests (Figure 7a and 8a), our Amphitheater can identify the important streams and assign higher priorities to them when the resource becomes scarce.

7.4 Effect of Virtual Seats

In the last part, we verify the effectiveness of virtual seat design in improving the efficiency of content dissemination. As we discussed in Section 4.1, the surrounding seat arrangement helps the dissemination of streams with content sharing and distribution. To verify these advantage brought by the virtual seats, we simulates two Amphitheaters. One with 500 audiences in the virtual seats, and one with zero audience. Other settings of this simulation are the same as the first part of the evaluation (Section 7.2).

Simulation Results. We plot the result in Figure 9, where the blue-diamond curve stands for the Amphitheater with audience sites and the red-square curve stands for the Amphitheater without audience sites. We can see from the figure that the performance is generally better when there are audience sites participating the session. Since the audience sites do not produce content, they play the role of hubs in the dissemination network. Recall that in our algorithm, a performer site will turn to the audience sites to receive requested streams when the other performer sites are saturated. Thus, when there are audience sites in the session, this mechanism lowers the request rejection ratio of performer sites (and hence increases the AQoS) because more senders are provided to receive the stream from.

8. DISCUSSION

Before we conclude the paper, there are some omitted issues and potential features that we would like to discuss about as some future directions.

Effect of Intensive View Changes. An issue to be addressed next is the effect of intensive view changes. For the design of the Amphitheater, each view change event of a site implies re-submission of its subscription request to the session management site; and the session management site would need to re-construct the dissemination forest. A naive reconstruction of the whole forest introduces computational (in the session management site) and communicational (session manager reassign senders to receivers) overheads. However, each view change of a viewer actually only affects its descendants in the dissemination tree she involves. This actually affects only part of the forest. Prior works [1, 9] have been investigating dissemination forest reconstruction schemes that can accommodate rapid view changes with prioritized evolutionary algorithm and site layering. The combination of these techniques with our Amphitheater is considered the next step of our development.

In-depth Investigation of the Virtual Seats. While the benefits brought by the virtual seat design are verified in this paper, we yet to deeply investigate the effect it has in the construction procedure of the forest. In our evaluation, we skipped the underlying characteristics of the dissemination trees (height, width, load balancing, etc.) constructed with and without the audience. Having a more thorough look at the effect it has on the forest may help us further improve the forest construction algorithm that we currently adopt.

Join and Leave of the Audiences. Changes in the audience are not considered yet in our work. At this point, a trivial way to handle a join event is to overlap the seat of the new comer with an existed audience. However, this will be a waste of the possibility of the new comer being an extra hub for content distribution if the seats can remain evenly dispersed. A leave event, on the other hand, will introduce an empty seat in the auditorium. This affects the underlying dissemination forest and has an effect similar to view change.

Changing Seats during the Performance. Although contradicting the sense in the real world, allowing seat changing during the session may be an incentive for users to choose a virtual theater over a real one. However, a negative effect of this feature is the loss of the benefit brought by the virtual seat design. Like the heavy-tail distribution of channel popularity in IPTV [14], very likely the audience will flock to the same area in the auditorium for a more interesting view point if we allow seat changing. The loss of even dispersion of the audiences may decrease the effectiveness of virtual seats on stream sharing.

9. CONCLUSION

In this paper we present the 3DTI Amphitheater, a live broadcasting system for dissemination of 3DTI content. We identify the hierarchical prioritization of streams in the construction of the content dissemination forest and test the system with real world network settings and configurations of real 3DTI system. Result shows that the Amphitheater outperforms prior 3DTI systems by boosting the AQoS while sustaining the same hundred-scale audience group.

10. ACKNOWLEDGMENTS

This material is based upon work supported by NSF Grant CNS10-12194, CNS09-64081KN.

11. REFERENCES

[1] A. Areifn, R. Rivas, and K. Nahrstedt, “Prioritized evolutionary optimization in open session management for 3D tele-immersion,” ACM MMSys, 2013

[2] W. Wu, Z. Yang, and I. Gupta et al., “Towards multi-site collaboration in 3D tele-immersive environments,” IEEE ICDCS, 2008

[3] G. Coulouris, J. Dollimore, and T. Kindberg et al., “Distributed systems: concepts and design,” Addison-Wesley, Fifth Edition, 2011

[4] X. Zhang, J. Liu, and B. Li et al. CoolStreaming/DONet: a data driven overlay network for peer-to-peer live media streaming,” IEEE INFOCOM, 2005

[5] Z. Yang, W. Wu, and K. Nahrstedt et al., “Enabling multi-party 3D tele-immersive environments with ViewCast,” ACM TOMCCAP, 2009

[6] P. Agarwal, R. Rivas, and W. Wu et al., “Bundle of streams: concept and evaluation in distributed interactive multimedia environments,” IEEE ISM, 2010

[7] Z. Yang, B. Yu, and K. Nahrstedt et al., “A multi-stream adaptation framework for bandwidth management in 3D Tele-immersion,” ACM NOSSDAV, 2006

[8] K. Nahrstedt, Z. Yang, and W. Wu et al., , “Next generation session management for 3D tele-immersive interactive environments,” Multimedia Tools and Applications, Springer Netherlands, 2011

[9] A. Arefin, Z. Huang, and K. Nahrstedt et al., “4D TeleCast: towards large scale multi-site and multi-view dissemination of 3DTI contents,” IEEE ICDCS, 2012

[10] Netmap. www.caida.org/tools/visualization/mapnet

[11] Teeve. cairo.cs.uiuc.edu/projects/teleimmersion/

[12] A. DeVincenzi, L. Yao, and H. Ishii et al., “Kinected conference: augmenting video imaging with calibrated depth and audio,” ACM CSCW, 2011

[13] H. Yin, X. Liu, and T. Zhan et al., “Design and deployment of a hybrid CDN-P2P system for live video streaming: experiences with LiveSky,” ACM MM, 2009
[14] V. Gopalakrishnan, R. Jana, and K. Ramakrishnan et al., “Understanding couch potatoes: measurement and modeling of interactive usage of IPTV at large scale,” ACM IMC, 2011
[15] M. Jain and C. Dovrolis, “End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput,” IEEE/ACM ToN, 2003
[16] D. Feldman and Y. Shavitt, “An optimal median calculation algorithm for estimating internet link delays from active measurements,” IEEE E2EMON, 2007
[17] Z. Wang and J. Crowcroft, “Quality of service routing for supporting multimedia applications,” IEEE JSAC, 1996
[18] P. Eugster, P. Felber, and R. Guerraoui et al., “The many faces of publish/subscribe,” ACM Computing Survey, 2003
[19] PPLive. http://www.pptv.com/
[20] Ustream. http://www.ustream.tv/
[21] Youtube. http://www.youtube.com/