SyncCast: Synchronized Dissemination in Multi-site Interactive 3D Tele-immersion

Zixia Huang, Wanmin Wu, Klara Nahrstedt, Raoul Rivas and Ahsan Arefin
Department of Computer Science
University of Illinois at Urbana-Champaign
{zhuang21, wwu23, klara, trivas, marefin2}@illinois.edu

ABSTRACT

An ideal interactive 3D tele-immersion (3DTI) system is expected to disseminate and synchronize multi-streams with a shortest-possible latency among participating sites, achieve *inter-stream synchronization*, and bound both *inter-sender* and *inter-receiver* skews. This is, however, a key challenge because of (1) the coexistence of multi-modal, correlated, bandwidth-savvy streams from multiple source media, (2) the bounded bandwidth resources at each site, (3) the heterogeneous transmission end-to-end delays (EED) between sites and (4) the diversity of 3D views requested by multiple users. Our study of the existing content dissemination topologies reveals their inadequacy of handlings the complication and dynamics present in 3DTI systems.

In this paper, we propose SyncCast, a multi-stream multicast dissemination scheme that takes into account the bandwidth constraint, as well as the inter-stream, inter-sender and inter-receiver synchronization. We classify the 3DTI media streams into different service classes based on the users' visual interests. SyncCast is designed to address the interactions among EED, synchronization and bandwidth in the real-world Internet settings. We compare SyncCast and our previous ViewCast algorithm [20]. The simulation results show the improvement in the synchronization performance and the implementation feasibility of SyncCast in supporting the multi-site interactive 3DTI system.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Video; C.2.1 [**Network Architecture and Design**]: Network communications

General Terms

Design, Performance, Experimentation

Keywords

SyncCast, 3D tele-immersion, synchronization

MMSys'11, February 23–25, 2011, San Jose, California, USA.

1. INTRODUCTION

Interactive 3D tele-immersion (3DTI) (Fig. 1) is an application in which geographically distributed users can achieve realistic collaborations in a joint virtual space. Some useful applications are medical consultation, remote education and collaborative entertainment. The characteristics of 3DTI include photorealistic video capturing of the participants, real-time dissemination of the media data, and multi-view rendering of the aggregated video. Users can thus interact seamlessly with remote peers in a joint virtual environment with auditory and visual feedback.

In a two-party interactive 3DTI system, synchronization of multiple sensory streams is already a big challenge. The system needs to synchronize the audio streams with multiple highly correlated video streams (e.g., the streams sent from site A to B in Fig. 2(a)). The correlation of video streams results from the deployment of 3D cameras in different positions around the stage to capture multiple views of the same scene. The timing synchronization of these different sensory streams is complicated by the fact that they have heterogeneous end-to-end delay (EED) between the *sender site* (i.e., the site which generates sensory streams) and the *receiver* site (i.e., the site which renders/plays sensory streams) [6] (Fig. 3). The reason is that different sensory streams employ their own protocols and adaptation algorithms in response to the bandwidth dynamics according to their diverse quality-of-service (QoS) requirements. The EED heterogeneity can impact the *inter-stream synchronization*. In addition, Internet dynamics (e.g. jitters) must be taken into account, and timely scheduling of audio playout and video rendering at the receivers thus becomes a key issue. Though this can be achieved by the receiver-side buffering, the buffer size can contribute to the overall EED, and a long EED can decrease the interactivity of the 3DTI system and may reduce the overall user satisfaction [15].

The extension of the 3DTI system for multi-site support creates increased complications to multi-stream dissemination and synchronization due to the following reasons.

(1) *Prolonged EED*: the possible relays though some *intermediate sites* (IS) (i.e., sites that relay media streams to other sites in Fig. 3[1]) in the application-layer overlay [20] can contribute to a prolonged EED between two sites. To improve the interactivity, a good 3DTI system should be able to minimize its *average system EED* (i.e., the average EED between all sender and receiver sites in the system). In addition, the aggregate jitter effects over multiple links can contribute to the latency variations which will further

[1]An intermediate site can also be a receiver site.

Figure 1: Collaborative 3DTI with 3 sites

impact the synchronization. To simplify the problem in this paper, we do not consider the jitter impacts and assume the fixed receiver-side buffer can smooth the Internet jitters. We use the equal buffer size for all receivers and do not include the buffer size in the EED computation.

(2) *Inter-receiver skew*: multiple receiver sites can experience different perceptions of the same 3DTI activity due to the heterogenous EEDs from a same sender site (of which the maximum EED difference is called *inter-receiver skew*). For example, in Fig. 2(a), site A is sending streams to all other sites. The maximum EED (A → C) is 180 msec while the minimum (A → B) is 20 msec, so the inter-receiver skew is 160 msec. The inter-receiver skew can cause unfairness when multiple people at different receiver sites require a timing privilege to conduct an activity. For example in a remote education application, students at different receiver sites (B, C, D) are racing to answer a question asked by an instructor at site A. In case of a double-talk when two or more people are talking at the same time, the third-party people at different receivers can also perceive the double-talks with different timing characteristics [14].

(3) *Inter-sender skew*: from an audience's perspective, some people in the 3DTI activity may be more distant or respond slower than others because of the heterogenous EEDs from multiple senders to a same receiver site (of which the maximum latency difference is called *inter-sender skew*). For instance in Fig. 2(b), site C is receiving the streams from A,B and D. The maximum EED (A → C) is 180 msec while the minimum (D → C) is 120 msec, so the skew is 60 msec. The inter-sender skew may lead to the misunderstanding of the audience located at the receiver site in a highly collaborative activity. An example is a collaborative basketball scenario, where a defensive player (site C in Fig. 2(b)) is interacting with two offensive players (site A and D). The temporal synchrony between player A and D can be seriously broken as perceived by player C if the inter-sender skew is large.

(4) *Inter-stream skew*: multi-streams from the same sender may follow different paths to reach the same receiver site [20]. The heterogeneity of EED incurred on multiple paths can further increase the *inter-stream skew*. For example in Fig. 2(a), site A has two streams s_1 and s_2 which follow two paths to site C. The incurred EED of s_1 is 180 msec while the latency of s_2 is 190 msec. So the skew is 10 msec. It is recommended in the 3DTI that the correlated video streams from the same site should be synchronized before they are sent to the renderer [6], and the audio-visual skew should fall within a 80-msec threshold [17] for lip synchronization. There has been a large body of work on the audio-visual synchronization [6, 11, 12, 17].

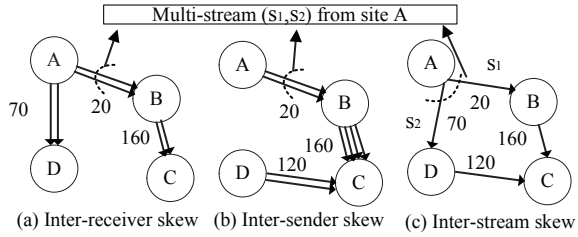

Figure 2: Synchronization Skews. Number presents the link latency (msec).

(5) *Bandwidth constraint*: Unlike the dissemination of the spatial audio [23] or VoIP, where data rate is usually small and audio signals can be mixed together when bandwidth is inadequate, the transmission of multiple 3D video streams can overload a site and 3D video mixing is often not accessible in real time due to the huge computation overhead.

(6) *Diversity of user interests*: multiple users can request different 3D video views and hence only a subset of video streams may be required [20] at each receiver site.

Note that even in an interactive 3DTI virtual environment (or a gaming system) with hundreds of active participants (or avatars), previous studies [22, 23] have shown that these active participants/avatars tend to communicate within a certain small-scale group, called *activity region*. Hence, it is more interesting and important to study the stream dissemination within a group of active participants/avatars.

Most of the previous studies working on application-layer centralized/distributed overlays or multicast trees are aimed at minimizing the delay or delay variation with or without bandwidth constraints [3, 7, 8, 13, 14, 23]. However they oversimplify the problem by either overlooking the combined impact of bandwidth overhead at each site caused by the coexistence of multiple overlays/trees for multi-stream dissemination of heterogeneous senders in the content dissemination topology, or by assuming the homogeneity of the media data requested at different sites. Therefore, their work cannot directly apply to our multi-site 3DTI system.

In our previous study, we have proposed ViewCast [20] algorithm which can effectively disseminate multiple video streams and allow cross-tree adjustment under bandwidth constraints and user view heterogeneity. But ViewCast does not take into account the interactivity and synchronization of the dissemination problem. [19, 2] have proposed multi-stream synchronization with bounded delay and delay variations. However, the two papers do not consider the constraints of the bandwidth and the inter-sender skew. In addition, [19, 2] only focus on the dissemination of video streams and do not take into account the audio's nature and impact on 3DTI synchronization.

In [6], we have also proposed a synchronization framework called TSync, which only operates under single-sender scenarios. TSync can bound the inter-stream skew within the recommended 80-msec threshold via the decision of the *dominant stream* (i.e., the view-central video stream in the *multi-stream bundle* [1] that is used for audio-visual synchronization), the implementation of timed synchronization points at multiple locations, the cooperative frame rate allocation for Internet bandwidth adaptation at the sender site, and the proper audio playout scheduling to synchronize to the 3D videos at the receiver site.

In this paper we propose SyncCast, a synchronized dissemination framework for multi-site interactive 3DTI with

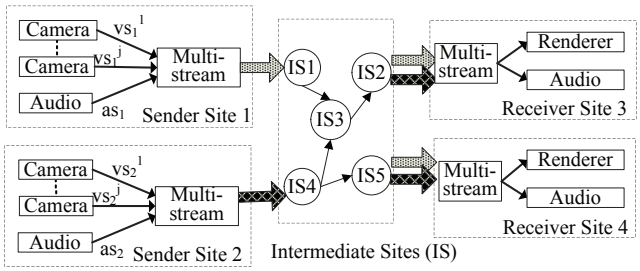

Figure 3: Multi-site 3DTI architecture.

multiple senders and receivers. The contributions are as follows. (1) We generalize the concept of synchronization in the multi-site interactive multimedia system, and model different synchronization aspects (inter-stream, inter-sender and inter-receiver synchronization) used in the real application. (2) We study the interactions of media data dissemination EED, synchronization and bandwidth overhead, and propose a multi-sender multi-receiver dissemination topology based on the interactions. The topology can minimize the average dissemination EED, bound inter-sender and inter-receiver skews, and guarantee inter-stream synchronization given the bandwidth constraint. (3) We take into account the diversity of user interests, and propose the concept of *global contribution factor* and *local contribution factor* which can describe the visual contribution of a certain video stream to all or only a subset of receiver sites respectively. We guarantee all receiver sites can receive the audio stream and the video streams containing the most important visual information relevant to the user view at each site respectively. (4) We bound the bandwidth overhead at each site either by re-selecting the alternative dissemination paths, or by applying cooperative bandwidth allocation at an overloaded site. (5) We evaluate SyncCast and several existing topologies based on the real Internet statistics collected in PlanetLab.

2. 3DTI SYSTEM MODEL

We present in this section the 3DTI system model (Fig. 3). A list of symbols and denotations is shown in Table 1. Throughout this paper, we use sites to represent nodes or vertices, and links to represent edges between two *neighboring sites* (i.e., sites that can connect each other).

2.1 Topology Model

We are given a set of N sites $(n_1, n_2, \ldots, n_N)$ in the 3DTI system. Each of the N sites can be classified as a sender or/and a receiver [2]. We let $\mathcal{E}^{snd}$ and $\mathcal{E}^{rcv}$ denote the set of sender and receiver sites in the system, and E_i^{snd}, E_k^{rcv} are the i-th sender site/k-th receiver site in $\mathcal{E}^{snd}$, $\mathcal{E}^{rcv}$. For each E_i^{snd}, it has a set of receiver sites $\mathcal{E}^{rcv(i)} = \{E_k^{rcv(i)}\}$, where $E_k^{rcv(i)}$ is the k-th receiver site in $\mathcal{E}^{rcv(i)}$. In this paper, we assume that for each sender site E_i^{snd}, all other sites are receiver sites of E_i^{snd} (i.e., $\mathcal{E}^{rcv(i)} = \{n_1, n_2, \ldots, n_N\} \backslash E_i^{snd}$) [3]. For each E_k^{rcv}, it has a set of sender sites $\mathcal{E}^{snd(k)} = \{E_i^{snd(k)}\}$, where $E_i^{snd(k)}$ is the i-th sender site in $\mathcal{E}^{snd(k)}$.

[2]A sender site can also be a receiver.

[3]A 3DTI sender site also receives and renders its own video streams. Because there is no transmission EED introduced over the Internet, we do not consider it as a receiver of its own streams in this paper.

Table 1: Notations and definitions

Notations	Definitions
n_i	The i-th participating site in 3DTI
$\mathcal{E}^{snd}$, $\mathcal{E}^{rcv}$	Set of sender or receiver sites in the system
E_i^{snd}, E_k^{rcv}	The i-th sender site or k-th receiver site
$\mathcal{E}^{snd(k)}$, $E_i^{snd(k)}$	Set of/i-th sender sites for E_k^{rcv}
$\mathcal{E}^{rcv(i)}$, $E_k^{rcv(i)}$	Set of/k-th receiver sites for E_i^{snd}
$L_{n_i \to n_k}$	One-way latency from n_i to n_k
$P_{i\to k}^s$	Dissemination path of stream s from E_i^{snd} to $E_k^{rcv(i)}$
$\mathcal{PS}_{i\to k}^s$	Set of all path options for stream s from E_i^{snd} to $E_k^{rcv(i)}$
$\text{EED}[P_{i\to k}^s]$	EED for $P_{i\to k}^s$
$B_{n_i}^{\text{in}}$, $B_{n_i}^{\text{out}}$	Incoming/outgoing bandwidth overhead at n_i
$\mathcal{S}_i$	Media stream set from E_i^{snd}
$\mathcal{AS}_i$, $\mathcal{VS}_i$	Audio or video stream set from E_i^{snd}
vs_i^j	j-th video stream from camera j of E_i^{snd}
as_i	Audio stream from E_i^{snd}
$DS_{i\to k}$	Video DS from E_i^{snd} to $E_k^{rcv(i)}$
$\mathcal{S}_{i\to k}^{NDS}$, $\mathcal{S}_{i\to k}^{NUS}$	Set of video NDS and NUS from E_i^{snd} to $E_k^{rcv(i)}$
$\mathcal{RS}_{i\to k}$	Set of all streams requested at $E_k^{rcv(i)}$ from E_i^{snd}
$\text{CF}_{i\to k}^s$	Contribution factor of stream s from E_i^{snd} at $E_k^{rcv(i)}$
$\Delta T_{i\to k}^{vs}$	Inter-video skew from E_i^{snd} to $E_k^{rcv(i)}$
$\Delta T_{i\to k}^{avs}$	Audio-visual sync skew from E_i^{snd} to $E_k^{rcv(i)}$
$\Delta T_{i\to k}^{s}$	Inter-stream skew from E_i^{snd} to $E_k^{rcv(i)}$
ΔT_k^{snd}	Inter-sender skew at E_k^{rcv}
ΔT_i^{rcv}	Inter-receiver skew of E_i^{snd}

We use the one-way latency $L_{n_i \to n_k}$ to describe the link cost between two neighboring sites. Note that there may not be a direct link between two sites (section 4.1). A path of a media stream s from E_i^{snd} to its receiver $E_k^{rcv(i)}$ is denoted as $P_{i\to k}^s$ [4]. $P_{i\to k}^s =< E_i^{snd}, n_1^{IS}, n_2^{IS}, \ldots, n_m^{IS}, E_k^{rcv(i)} >$, may be routed through several IS (i.e., $n_1^{IS}, n_2^{IS}, \ldots, n_m^{IS}$) in the application-layer overlay or multicast tree. The total cost of the path $P_{i\to k}^s$ (i.e., EED) is denoted as $\text{EED}[P_{i\to k}^s]$. $\text{EED}[P_{i\to k}^s]$ can be computed by summing up all the link costs on the path and the processing latencies incurred on sender, intermediate and receiver sites. In this paper, we assume all the processing latencies are negligible. Note that there can be be multiple path options from E_i^{snd} to $E_k^{rcv(i)}$ for stream s, and we denote the set of path options $\mathcal{PS}_{i\to k}^s$.

Each site n_i has an in-degree and out-degree bandwidth upper bound (i.e., $\max B_{n_i}^{\text{in}}$ and $\max B_{n_i}^{\text{out}}$) to constrain the bandwidth overhead $B_{n_i}^{\text{in}}$ and $B_{n_i}^{\text{out}}$. We assume the bandwidth consumption of the audio streams is negligible compared to the videos.

2.2 Stream Model

As Fig. 3 shows, each sender site E_i^{snd} contains M 3D cameras and one audio microphone component. The M cameras at E_i^{snd} produce a video *stream bundle* $\mathcal{VS}_i = \{vs_i^j\}$ [1] with stream vs_i^j from camera j. Each vs_i^j consists of a sequence of frames $\{f_i^j(t)\}$ captured at time t. $\mathcal{VS}_i$ can also be represented as a sequence of video *macroframes* $mf_i(t)$ (i.e., $\mathcal{VS}_i = \{mf_i(t)\}$), where a macroframe $mf_i(t)$ is a set of video frames taken from different cameras at E_i^{snd} at the same time t, i.e., $mf_i(t) = \{f_i^1(t), \ldots, f_i^M(t)\}$. Each audio microphone can produce one audio stream $\mathcal{AS}_i = as_i$

[4]Throughout this paper, $i \to k$ denotes E_i^{snd} to $E_k^{rcv(i)}$.

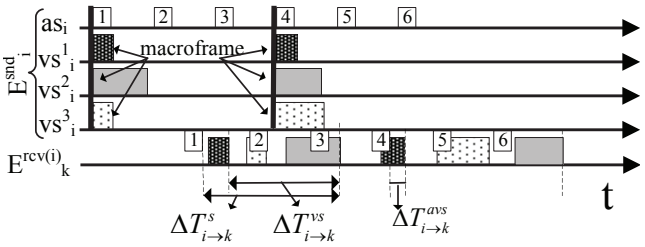

Figure 4: Inter-stream skew, Inter-video skew, and audio-visual skew from E_i^{snd} to $E_k^{rcv(i)}$. vs_i^1 is the DS.

(i.e., $\|\mathcal{AS}_i\| = 1$) consisting of periodic audio frames [14]. The set of all sensory streams from E_i^{snd} are denoted as $\mathcal{S}_i = \{\mathcal{AS}_i, \mathcal{VS}_i\}$.

Each receiver site $E_k^{rcv(i)}$ contains one renderer (display)[5] and one audio speaker component. We define the concept of a *dominant stream* (DS) $DS_{i\to k}$ within a bundle $\mathcal{VS}_i$ to be the most important video stream for the display from E_i^{snd}. It is the stream with the maximum *contributing factor* (CF) which represents the contribution of the stream's 3D image pixels to the user view. CF is determined using (a) camera orientation of a video stream $s = vs_i^j$, $\vec{O}_s$, and (b) the desired user view orientation $\vec{O}_{u_k^i}$ at $E_k^{rcv(i)}$ from site E_i^{snd} [21]: $\text{CF}_{i\to k}^s = \vec{O}_s \cdot \vec{O}_{u_k^i}$. Other video streams in the correlated 3D video bundle that are not DS but satisfy $\text{CF} > \text{CF}_0$ are called *non-dominant streams* (NDS), where CF_0 is the lower bound of CF which has the visual contribution [20]. Streams whose CF is less than CF_0 are called *non-use streams* (NUS). A receiver site E_k^{rcv} may only request a subset in the video stream bundle from E_i^{snd}, and this subset is denoted as $\mathcal{RS}_{i\to k}$. Note that the concept of $\mathcal{RS}$ also extends to the audio stream and in this paper, all receiver sites always request the audio.

2.3 Synchronization Model

Multiple video streams from the same sender can arrive at a receiver at different time. We define the *inter-video skew* to denote this difference. The skew $\Delta T_{i\to k}^{vs}$ of the video stream from E_i^{snd} to $E_k^{rcv(i)}$ can be computed by:

$$\Delta T_{i\to k}^{vs} = \max_{s,\ s'\in\mathcal{RS}_{i\to k}\setminus as_i} |\text{EED}_{i\to k}^{s} - \text{EED}_{i\to k}^{s'}| \quad (s \neq s') \qquad (1)$$

Because it is impossible to synchronize the audio to all video streams in the bundle, we also define the *audio-visual skew* $\Delta T_{i\to k}^{avs}$ to describe the EED difference between the audio and video DS from the same sender site, i.e.,

$$\Delta T_{i\to k}^{avs} = |\text{EED}_{i\to k}^{as} - \text{EED}_{i\to k}^{DS_{i\to k}}| \qquad (2)$$

The audio-visual skew is important because it can cause lip asynchrony which will degrade human subjective perceptions. We compare the audio with the video DS because video DS contains the most important visual information.

The inter-stream skew, on the other hand, is defined as the maximum EED difference among all audio and video streams from a sender site to a receiver. It is denoted as $\Delta T_{i\to k}^{s}$.

$$\Delta T_{i\to k}^{s} = \max_{s,\ s'\in\mathcal{RS}_{i\to k}} |\text{EED}_{i\to k}^{s} - \text{EED}_{i\to k}^{s'}| \quad (s \neq s') \qquad (3)$$

Fig. 4 shows the relationship between $\Delta T_{i\to k}^{s}$, $\Delta T_{i\to k}^{vs}$ and $\Delta T_{i\to k}^{avs}$. Note that $\Delta T_{i\to k}^{vs} \leq \Delta T_{i\to k}^{s}$.

In order to facilitate audio-visual synchronization, in this paper we prescribe that the audio stream as_i from sender site E_i^{snd} and its corresponding video DS ($DS_{i\to k}$) to a receiver site $E_k^{rcv(i)}$ follow the same transmission path to minimize the EED difference. We call the audio stream and video DS *audio-visual dominant stream* (AVDS).

[5]We assume there is only one display at each receiver site.

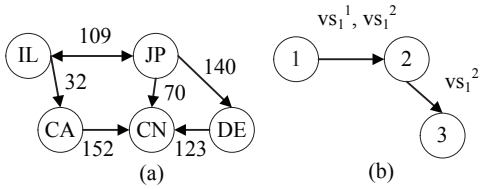

Figure 5: (a) Illustration of problem formulation, numbers are delays in msec. (b) Unfair routing.

The inter-receiver skew ΔT_i^{rcv} of E_i^{snd} is defined as the difference in EED of the video DS to multiple receiver sites, i.e.,

$$\Delta T_i^{rcv} = \max_{k,\ k'\in\mathcal{E}^{rcv(i)}} |\text{EED}_{i\to k}^{DS_{i\to k}} - \text{EED}_{i\to k'}^{DS_{i\to k'}}| \quad (k \neq k') \qquad (4)$$

The inter-sender skew ΔT_k^{snd} of E_k^{rcv} is defined as the difference in EED of the video DS from multiple sender sites.

$$\Delta T_k^{snd} = \max_{i,\ i'\in\mathcal{E}^{snd(k)}} |\text{EED}_{i\to E_k^{rcv}}^{DS_{i\to E_k^{rcv}}} - \text{EED}_{i'\to E_k^{rcv}}^{DS_{i'\to E_k^{rcv}}}| \quad (i \neq i') \qquad (5)$$

There are several things to note here. First, due to the space limit, we assume that all receiver sites are using the fixed jitter buffer with the same buffer size for all media streams, so we do not take into account the impact of jitters on synchronization skews. Second, we compute the inter-sender and inter-receiver skews based on the timing information of the AVDS, because AVDS contains the most important audio-visual information, and is thus given top priority for dissemination in our paper (section 5). Most of the receiver sites can receive multiple AVDS from different sender sites, and can play the 3D videos with high quality even though they do not receive any video NDS. Last, only the AVDS's arrival timing information at the receiver affects the audio playout and 3D multi-view rendering time, which is the deterministic factor deciding the interactivity of the multi-site 3DTI. We will show in section 5 that inter-stream and inter-video skews do not affect the multi-site interactions using our algorithm.

3. PROBLEM FORMULATION

As discussed in Section 1, the goal of our paper is to develop an interactive multi-site 3DTI system which can minimize the average system EED under both synchronization and bandwidth constraints, and each receiver site should be given the top priority to receive the AVDS in order to guarantee the audio-visual rendering quality.

An illustration of our problem is shown in Fig. 5(a) where IL and JP are the sender sites, and all sites are receivers (sender sites do not receive streams from themselves). There are a total of 8 sender-receiver pairs in the graph. Each sender site has two streams (i.e., s_{IL}^1, s_{IL}^2, s_{JP}^1, s_{JP}^2). Each receiver site requests one DS and one NDS from each sender site. We assume that s_{IL}^1 and s_{JP}^1 are DS for all receivers, and the other two streams are NDS. For each sender-receiver pair, there can be one or multiple path options. For example, between IL and CN there are three path options for all streams $\mathcal{PS}_{\text{IL}\to\text{CN}}[0] = <\text{IL,JP,CN}>$, $\mathcal{PS}_{\text{IL}\to\text{CN}}[1] = <\text{IL,CA,CN}>$ and $\mathcal{PS}_{\text{IL}\to\text{CN}}[2] = <\text{IL,JP,DE,CN}>$, and different paths lead to different costs (EEDs). Our goal is to

select a path, for each stream of each sender-receiver pair, from a list of path options, and to combine the selected paths of all the sender-receiver pairs into an dissemination overlay such that (1) in the sub-overlay disseminating only AVDS, the resulting average system EED is minimized; (2) synchronization metrics (inter-sender, inter-receiver and inter-stream skews) and incoming/outgoing bandwidth overhead are bounded.

The problem can be formulated in a general form as follows. δ_s, δ_{src} and δ_{rcv} are the upper bounds of inter-stream, inter-sender and inter-receiver synchronization.

• **Minimizing average system EED of AVDS paths:**

$$\textbf{Min } \frac{1}{\sum_i^{|\mathcal{E}^{snd}|} |\mathcal{E}^{rcv(i)}|} \sum_{E_i^{snd}} \sum_{E_k^{rcv(i)}} \text{EED}[P_{i\to k}^{DS_{i\to k}}]$$

• **Synchronization constraint :**

$$\forall\ E_i^{snd}, E_k^{rcv}: \qquad \Delta T_{i\to k}^{vs} \le \Delta T_{i\to k}^{s} < \delta_s$$
$$\Delta T_k^{snd} < \delta_{snd}, \quad \Delta T_i^{rcv} < \delta_{rcv}$$

• **Bandwidth constraint :**

$$\forall\ n_i:\ B_{n_i}^{\text{in}} < \max B_{n_i}^{\text{in}}, \quad B_{n_i}^{\text{out}} < \max B_{n_i}^{\text{out}}$$

Note that because of bandwidth and synchronization constraints, some receiver sites may not receive the AVDS from one or multiple sender sites. We call these receiver sites *victim sites*, which are sites that cannot receive at least one AVDS. In this situation, we are aimed at first maximizing the number of *successfully*-received AVDS and given the successful AVDS, then minimizing their average system EED.

In a multi-site 3DTI system, there are several application-specific factors that affect the construction and optimization of the content dissemination scheme.

First, due to the diversity of user views, different receivers may request different video DS and NDS. Previous studies [3, 7, 8, 13] assume the multi-stream homogeneity from the same sender, and hence it is possible that some IS may have to relay some video streams which are of no use (i.e., NUS) to them. For example, in Fig. 5(b), site 2 has to relay vs_1^2 even though it does not request the stream. This will cause an *unfair routing* in the transmission topology. A good scheme should be able to achieve *fair routing* in which the IS only relay streams that they themselves may request.

Second, in a dissemination scheme with multiple multicast trees, cross-tree adjustment [10, 20] is not the only feasible solution to satisfy the bandwidth constraint. The characteristics of 3DTI multi-streams can allow adaptations at an overloaded site by cooperative bandwidth allocation to more important streams and achieve frame rate allocation [6]. However, it may degrade the overall visual quality due to the motion jerkiness due to the reduced frame rate.

To sum up, the overall problem can be divided into two parts. The first part only considers multi-stream homogeneity, and minimize average system EED given the synchronization and bandwidth constraints. We use the real link costs by collecting the one-way latencies between different sites in the real Internet settings (section 4). There have been numerous existing studies constructing dissemination topologies aimed at minimizing delay related metrics [3, 7, 8, 13]. We evaluate these studies in section 4. The second part focuses on 3DTI-specific multi-stream heterogeneity in deciding the topology. We will consider both parts in designing our own algorithm (section 5).

4. EVALUATION OF DELAY-SENSITIVE TOPOLOGIES

Previous studies on delay-sensitive topologies [3, 7, 13, 19, 23] conducted evaluations based upon link costs which are randomly selected within a certain range, computed by propagation delays in a wireless setting, or obtained in the virtual gaming environment. Due to the delay correlations over the Internet, their results can neither describe a real network environment for multi-site 3DTI, nor can they present real EED heterogeneity, synchronization behaviors, bandwidth overhead and their interactions. In this section, we collect the one-way latency statistics from PlanetLab to simulate a real Internet setting. We construct dissemination topologies based on real link costs between sites, and study the interactions of EED, synchronization and bandwidth overhead resulting from the Internet delay distributions. Based on their timing and bandwidth consumption characteristics, we design our own algorithm in section 5.

4.1 Network Environment

We present the one-way latencies (link costs) collected from PlanetLab in Table 2 and 3 to simulate both 5-site and 9-site real 3DTI settings. We sent UDP packets from each site to all others at the same time and computed the one-way latency by halving the round-trip time information. The size of the packets was smaller than the maximum transmission unit (MTU) in order to avoid fragmentation. Because different sites can exhibit different latency variations, we consider both 5-site and 9-site communications and classify the network environment into three scenarios based on their geographical locations: (1) all sites in US (represented as 5A and 9A), (2) sites in both US and Europe (5B and 9B), and (3) sites in US, Europe and Asia (5C and 9C). In addition, widely-deployed firewall and poor link conditions (high loss rate, large jitter or low data rate) can impede the direct connections and transmissions of real-time traffic between certain sites. Hence, we remove some of the links and represent the new network environment as 5D-5F and 9D-9F (shown in the grey boxes in Table 2 and 3) to simulate the firewall-blocked connection (e.g. the link between IL and IN in Table 2(a)) or poor link conditions (e.g., links between Beijing and all sites outside China in Table 3(c)). So there are total of 12 environment for evaluation.

Table 2 and 3 show the correlations of one-way latency in the real Internet environment. First, sites that are close in the geographical locations usually exhibit similar latency distributions. For example, IL1 and IL2 in Table 3 have similar one-way latencies to other sites. Second, there are correlations among the links shared by a same site. For instance, the delays from BJ to all sites outside China are greater than 180 msec possibly due to the poor International connections there. We will show in section 4.3 the impact of latency correlations on the existing content dissemination schemes.

4.2 Classification of Content Dissemination Schemes

We classify the existing content dissemination schemes into four categories. These schemes are designed not for specific applications and do not take into account the heterogeneity within the multi-stream bundle (i.e., inter-stream synchronization skew).

(1) A *full-mesh scheme* requires each sender site to send

Table 2: One-way delay (msec) for 5 nodes. (a) in US, (b) in US and Europe, (c) in US, Europe and Asia.

Table 3(a): 5 nodes in US (5A,5D)

5A	CA	IL	FL	IN	NY
CA	0	32	35	33	43
IL	32	0	28	10	21
FL	35	28	0	24	24
IN	33	10	24	0	18
NY	43	21	24	18	0

Table 3(b): 5 nodes in US and Europe (5B,5E)

5B	CA	IL	UK	DE	NY
CA	0	32	80	88	43
IL	32	0	72	75	21
UK	80	72	0	18	53
DE	88	75	18	0	52
NY	43	21	53	52	0

Table 3(c): 5 nodes in US, Europe and Asia (5C,5F)

5C	CA	IL	DE	JP	CN
CA	0	32	88	70	152
IL	32	0	75	109	178
DE	88	75	0	140	123
JP	70	109	140	0	34
CN	152	178	123	34	0

multiple streams directly to the receiver sites via unicasts. This scheme usually results in the small EED, inter-sender and inter-receiver synchronization skews, but it is bottlenecked at the bandwidth availability of each sender. In addition, because of the firewall-blocked connections and poor link conditions, a full-mesh scheme may not always be accessible in the 3DTI system.

(2) A *centralized scheme* requires an intermediate dedicated server/site to relay the multi-streams from the senders to the receivers. This scheme is usually adopted in the multi-party VoIP (e.g., Skype[6] and QQ[7]) due to the low bandwidth demand of the audio streams and the feasibility of mixing audio signals. Although it increases the EED compared to a full-mesh topology, a centralized scheme can avoid the blocked connection due to the firewall deployment or prevent transmissions over poor links. The 3DTI however cannot use this scheme due to the huge network and computation demand of video streaming and processing.

(3) An extension of the centralized scheme is a *hybrid scheme* (usually an overlay network) in which there can be more than one IS to relay the media data. Each site communicates with the nearest IS in the overlay. One example of a hybrid scheme is proposed in [14]. Compared to a centralized scheme, the increased number of IS can reduce the chances of bandwidth overload. But it further increases the EED and can degrade both inter-sender and inter-receiver synchronization. The construction of an overlay can vary depending on different optimization goals. For example, the algorithm in [14] is aimed at minimizing the maximum EED in the overlay.

(4) An application-layer *multicast tree* in essence is another type of overlay in which the number of IS is usually not bounded or preset, and multiple streams from the same/different sender sites may follow different trees. The coexistence of multiple trees rooted at the same/different senders forms a multicast forest. The EED, inter-sender and inter-receiver skews can vary depending on tree construction algorithms. Multiple trees used for multi-stream dissemination from the same sender can create a multi-stream skew. Given the bandwidth constraint at each site, the tree construction algorithms can be generally divided into three layers. The bottom layer is the multicast tree construction with certain optimization goals (e.g., reducing the average EED and/or bounding EED variations, etc.). Some examples are the minimum spanning tree (MST) [8], minimal diameter spanning tree (MDST) [4], the shortest path tree (SPT) [23] or the DVMA algorithm with bounded delay and delay variations [13]. The medium layer is the bandwidth adjustment within the tree. For each tree constructed in the bottom layer, the algorithms [10, 20] reduce the overloaded sites either by pruning (sacrificing some victims) or by intra-tree adjustment achieved via removing some descendants of the overloaded sites to those that have more bandwidth availability. However, these problems have been proven to be NP complex [9, 18]. Hence, most of those adjustment algorithms are quite heuristic and their effectiveness can vary, depending on specific applications. In addition, adjustment can be difficult for a tree with a large number of sites, so there are some algorithms that combine the bottom and medium layers together into a single layer and directly construct a multicast tree with a bounded incoming/outgoing degree [16]. The top layer is the bandwidth adjustment across the trees (within a forest) [10, 20], because multiple trees can share some sites which may cause the bandwidth overload at these sites. Just as the medium layer, adaptation across multiple trees can also vary at different applications.

[6]Skype: `http://www.skype.com`
[7]QQ: `http://www.qq.com`

Table 3: One-way delay (msec) for 9 nodes. (a) in US, (b) in US and Europe, (c) in US, Europe and China.

Table 4(a): 9 nodes in US (9A,9D)

9A	CA1	CA2	IL1	IL2	IL3	FL	IN	NY	TX
CA1	0	2	32	31	32	35	33	43	18
CA2	2	0	31	35	32	35	20	46	19
IL1	32	31	0	1	2	28	10	21	19
IL2	31	35	1	0	1	28	11	21	19
IL3	30	32	2	1	0	28	11	21	20
FL	35	35	28	28	28	0	24	24	27
IN	33	20	10	11	11	24	0	18	21
NY	43	46	21	21	21	24	18	0	31
TX	18	19	19	19	20	27	21	31	0

Table 4(b): 9 nodes in US and Europe (9B,9E)

9B	CA	IL1	IL2	FL	IN	UK	DE1	DE2	IT
CA	0	32	31	35	33	80	88	90	102
IL1	32	0	1	28	10	72	75	81	95
IL2	31	1	0	28	11	74	72	85	98
FL	35	28	28	0	24	85	86	82	92
IN	33	10	11	24	0	70	71	76	93
UK	80	72	74	85	70	0	18	18	24
DE1	88	75	72	86	71	18	0	6	17
DE2	90	81	85	82	76	18	6	0	21
IT	102	95	98	92	93	24	17	21	0

Table 4(c): 9 nodes in US, Europe and China (9C,9F)

9C	CA	IL1	IL2	UK	DE1	DE2	SH	BJ	HK
CA	0	32	31	80	88	90	152	193	167
IL1	32	0	1	72	75	81	178	204	179
IL2	31	1	0	74	72	85	175	203	182
UK	80	72	74	0	18	18	155	187	160
DE1	88	75	72	18	0	6	143	181	152
DE2	90	81	85	18	6	0	140	180	153
SH	152	178	175	155	143	140	0	61	55
BJ	193	204	203	187	181	180	61	0	87
HK	167	179	182	160	152	153	55	87	0

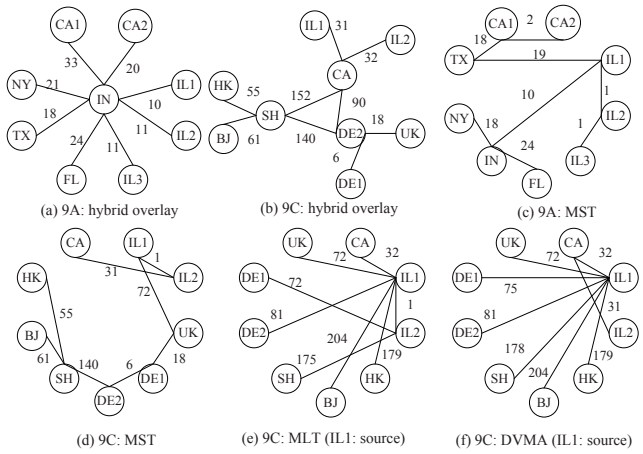

Figure 6: Resulting content distribution topologies based on the Internet data in Table 2 and 3. Number by the links are the one-way latency (in msec).

4.3 Evaluation of Five Topologies

We evaluate the following five topologies typically used in content dissemination. In this section, we assume all sites in the topology are both senders and receivers. We do not consider factors that are specific to 3DTI applications.

(1) Full-mesh topology.

(2) Hybrid overlay [14] which uses a greedy approach to increase the number of intermediate sites in order to minimize the maximum EED in the overlay.

(3) Minimum spanning tree (MST) based on Kruskal's Algorithm [8] which finds the cheapest unmarked edges (links) that does not form a marked loop.

(4) Minimum latency tree (MLT) [23] which is constructed by combining all shortest paths from the senders to their receiver sites using the Dijkstra's Algorithm.

(5) Delay variation multicast algorithm (DVMA) [13] which constructs a tree with bounded delay and delay variation based on k-shortest paths from the sender sites to the receivers.

Table 4 presents the simulation results for the five topologies under network environment 5A-5F and 9A-9F. Note that both full-mesh network and hybrid overlay are not available under network environment 5D-5F and 9D-9F because they require the link latency information between every two neighboring sites in the topology. Several observations are presented as follows.

EED. Table 4 shows that MST leads to the largest average and maximum EED, due to the largest number of IS introduced in the topology (e.g., four IS on the path from CA2 to FL in Fig. 6(c)). The EED can be further increased if the processing latencies on these IS are not negligible. On the contrary, all other topologies can achieve relatively small EED, of which MLT leads to the minimum EED.

Inter-sender and inter-receiver skews. Due to the EED variations across different sites, MST results in the largest inter-sender and inter-receiver skews. The EED variations in the full-mesh, hybrid overlay and MLT are comparatively small, but still their topologies may exceed the upper bound of both synchronization skews (say, 200 msec). DVMA is constructed based on MLT by keeping the path with the largest EED in the topology and replace the shortest path with other alternatives in order to reduce the synchronization skews. E.g., in Fig. 6(f), IL1 sends media data to IL2 via CA instead of direct communication as in MLT in Fig. 6(e). This can increase the latency between IL1 and IL2 from 1 msec to 63 msec, and thus reduce the maximum synchronization skew from 203 msec to 175 msec. Note that DVMA may not always be effective if there is no alternative path (e.g. the connection between IL1 and IL2 network environment 9F in Table 4(b)).

Bandwidth overhead. Bandwidth availability is an issue in all algorithms. In a full-mesh topology, each sender site needs to send multiple copies of media streams to its receivers. A hybrid overlay can lead to a centralized topology which will overwhelm the central site when delay variation is small (e.g., network environment 9A in Fig. 6(a)). In MST, many sites (e.g., UK, DE1, and etc. in Fig. 6(d)) need to relay all streams to other sites even though they only need a subset of these streams. MLT and DVMA can create multiple trees in the topology rooted at different sender sites, but none of them considers the bandwidth constraints within/across trees (e.g., site IL1 in Fig. 6(e) and (f)).

Interactions of different metrics. Those network-layer quality metrics exhibit strong interactions. First, the topology may not lead to the smallest average system EED if we want to reduce the synchronization skews by replacing the paths with the shortest EED with k-shortest alternatives (as in DVMA), or to satisfy the bandwidth constraints by preventing the stream dissemination being routed through the shortest paths. Second, due to the Internet delay correlations, media data from a sender site can be routed through a geographically close intermediate sites for the shortest paths (e.g., IL1 uses IL2 as an intermediate site in Fig. 6(e)). If IL2 is not a sender site, the routing paths via IL2 can effectively reduce the bandwidth overhead at IL1. However, IL1 will use the bandwidth resources at IL2 if IL2 itself is a sender site.

Of all the five topologies, DVMA can lead to a topology with the smallest EED given the constraint of the inter-receiver skew compared to other algorithms. Unfortunately, none of the existing topologies has studied the interactions among EED, synchronization and bandwidth metrics in a real Internet setting. Nor do these studies consider the heterogeneity within a multi-stream bundle. Hence, based upon DVMA, we design SyncCast to address these issues in section 5.

5. DESIGN OF SYNCCAST

We design SyncCast based on the multi-stream heterogeneity and the interactions of EED, synchronization and bandwidth metrics in the real Internet settings. We adopt a multi-tree structured scheme rooted at different sender sites because of its flexibility to both Internet dynamics and synchronization/bandwidth adaptations. We compare SyncCast with ViewCast [20] during our discussions. We study the design of receiver-side buffer and its impact on synchronization in SyncCast at the end of this section.

5.1 Design Rationale

This type of optimization problems has been proven to be NP hard [9, 18], and an optimal solution is inaccessible without introducing some heuristics. Here, we discuss the rationale behind the design of our SyncCast algorithm.

1. **Stream prioritization**. We prioritize multi-streams from different sender sites because of the limited amount

Table 4: Minimum and maximum inter-sender/inter-receiver skew (msec) as well as average and maximum EED (msec) for the 5 topologies under 12 network environment (5A-5F,9A-9F in Table 2 and 3).

Table 4(a)			Full-mesh	Hybrid	MST	MLT	DVMA
5A	Skew	min	11	23	24	11	11
		max	25	33	42	25	25
	EED	avg	26.8	34	35.6	26.8	26.8
		max	43	57	66	43	43
5B	Skew	min	32	31	49	32	32
		max	70	53	105	70	70
	EED	avg	53.4	67.6	63.8	53.4	53.4
		max	88	105	123	88	88
5C	Skew	min	65	88	75	65	65
		max	146	152	177	106	106
	EED	avg	100.1	135.8	104.8	90.4	90.4
		max	178	240	211	140	140
9A	Skew	min	11	23	35	11	11
		max	44	33	71	43	43
	EED	avg	23.1	32.9	31.9	22.5	22.5
		max	46	57	73	45	45
9B	Skew	min	67	84	95	67	67
		max	97	103	130	95	95
	EED	avg	55.7	69.4	70.3	55.4	55.4
		max	102	127	147	102	102
9C	Skew	min	123	129	195	121	123
		max	203	213	298	203	175
	EED	avg	114.2	140.6	152.4	113.7	115.5
		max	204	245	329	204	204

Table 4(b)			MST	MLT	DVMA
5D	Skew	min	35	11	11
		max	71	41	41
	EED	avg	45.8	33.3	33.3
		max	95	59	59
5E	Skew	min	49	32	32
		max	105	86	86
	EED	avg	63.8	56.8	56.8
		max	123	104	104
5F	Skew	min	75	65	65
		max	177	106	106
	EED	avg	104.8	90.4	90.4
		max	211	140	140
9D	Skew	min	42	12	12
		max	100	50	50
	EED	avg	40.4	27.2	27.2
		max	102	52	52
9E	Skew	min	81	64	64
		max	172	82	82
	EED	avg	87.7	59.2	59.2
		max	189	104	104
9F	Skew	min	195	124	124
		max	297	239	239
	EED	avg	152.4	119.5	119.5
		max	329	240	240

of network resources. The dissemination of 3DTI streams can be divided into three service classes: (1) AVDS from the sender sites to their corresponding receivers; (2) streams that have been disseminated as AVDS, but can also serve as NDS for other receivers; and (3) streams that have never been serviced, and can only serve as NDS for all receiver sites (sole-NDS). In our design, we give top priority to all AVDS streams, and last priority to the sole-NDS.

Streams within the same class can also be prioritized. We propose two concepts based on the tree-structured multicast topology: *global contribution factor* (GCF) and *local contribution factor* (LCF). The GCF of a stream s in the the j-th class (GCF_j^s) is defined as the accumulative CF to all designated receiver sites within the class. A formal definition is as follows.

$$\begin{aligned}
\text{GCF}_1^s &= \sum_{i,k} CF_{i\to k}^s && s = DS_{i\to k} \\
\text{GCF}_2^s &= \sum_{i,k} CF_{i\to k}^s && s \in \mathcal{S}_{i\to k}^{NDS} \ \&\ \exists\ k' \neq k, s = DS_{i\to k'} \\
\text{GCF}_3^s &= \sum_{i,k} CF_{i\to k}^s && s \in \mathcal{S}_{i\to k}^{NDS} \ \&\ \nexists\ k' \neq k, s = DS_{i\to k'}
\end{aligned} \tag{6}$$

The metrics GCF describes a stream's overall contribution to all receiver sites in each stream class. A stream with a larger GCF is more visually important to whole system compared to other streams within the same class.

The LCF of a stream s at a node n_i ($\text{LCF}_{n_i}^s$), on the other hand, is defined as the accumulative CF for all n_i's descendants in the tree (i.e., nodes that receive s through the relay of n_i). The metrics LCF describes the overall contribution of relaying each stream at an intermediate node. A stream with a larger LCF at a node should be given higher priority.

In comparison, ViewCast does not prioritize streams within the same service class. Its prioritization is only based on the CF and it uses preemption for service differentiation. CF alone however can only tell the contribution of a stream to a single receiver, and does not describe its importance to the whole system. This can create an issue when two streams (one with a small CF and a large GCF, and the other with a large CF and a small GCF) competing for the limited resources. ViewCast in this case will prefer the second stream, and thus can increase the likelihood of victim sites and *victim streams* (i.e., the number of DS/NDS not received).

2. **Path prioritization**. A stream from a sender site to a receiver can follow multiple paths. Both EED and fairness impact the path prioritization. To evaluate the fairness of a path, we compute on each path $\mathcal{PS}_{i\to k}^s[j]$ (i.e., the j-th path in path options $\mathcal{PS}_{i\to k}^s$ for stream s) the number of IS where s has no contributions, and denote the number as $Q[j]$. Suppose $\mathcal{PS}_{i\to k}^s[j]$ has M IS, i.e., $\mathcal{PS}_{i\to k}^s[j] = \{E_i^{snd}, n_1^{IS}, n_2^{IS}, \ldots, n_M^{IS}, E_k^{rcv(i)}\}$, $Q[j]$ can be computed by

$$Q[j] = \sum_{m=1}^{M} I\ \{CF_{E_i^{snd}\to n_m^{IS}}^s \leq \text{CF}_0\} \tag{7}$$

Here $I\{\cdot\}$ is the indicator function.

To improve the fairness of the dissemination topology and prevent nodes relay NUS which otherwise would waste their own network resources, while preserving the interactivity, we prioritize path candidates based on a *fairness-first, latency-next* policy in our system, meaning that the paths with a smaller $Q[j]$ are always placed at a higher priority, and that the path with a $Q[j]$ and a shorter EED is better than those with the same $Q[j]$ and a longer EED.

By contrast, fairness is also guaranteed in ViewCast because the receiver sites only request a stream from the sender site and those that have already requested and received the streams. However, multiple DS can compete for bandwidth resources at a site, and thus may cause a degradation of audio/visual quality in ViewCast.

3. **Bandwidth preservation policy**. The 3DTI system has a very high demand of bandwidth for multi-stream dissemination. To provide the shortest dissemination paths in the real Internet setting and offer resilience to the user view changes, we prescribe in our system that: unless a site needs to relay a stream which is a DS for other sender-receiver pair and their is no alternative path available for this DS,

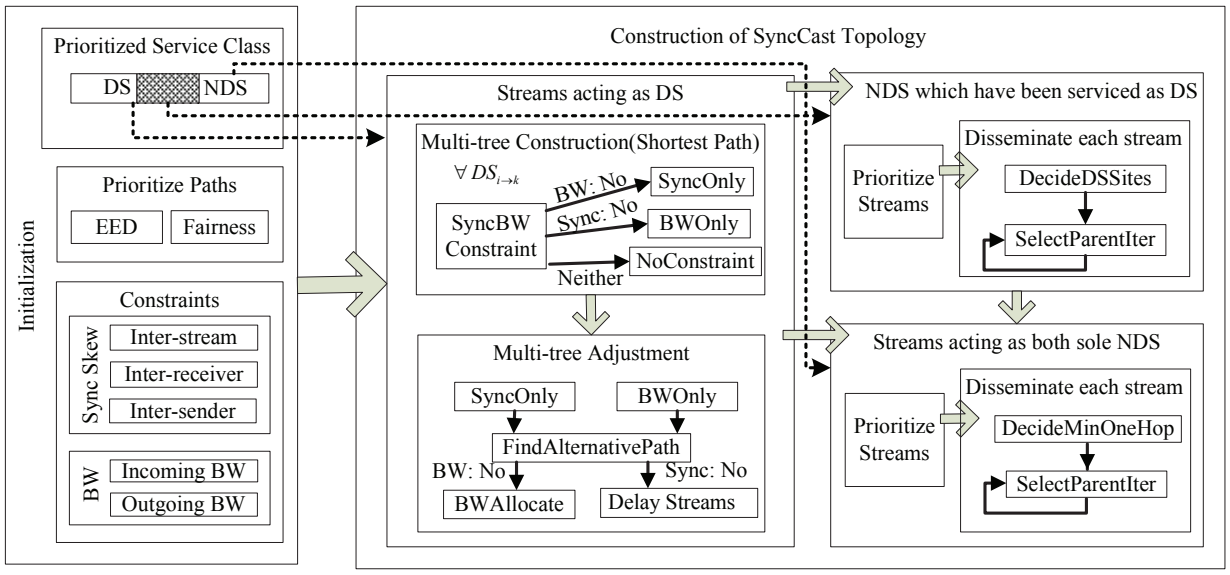

Figure 7: An overview of SyncCast.

• no site is responsible for relaying its NUS (which can be DS/NDS to other receivers) from the senders;
• no sender site should relay its NDS from other senders.

The ViewCast algorithm does not have this layer of bandwidth preservation because of a lack in the service class differentiation.

5.2 SyncCast Design Details

Based on the design rationale, we propose SyncCast. Fig. 7 shows its framework. We use GCF to prioritize the streams within each service class.

5.2.1 An Overview

SyncCast first determines the dissemination topology for all AVDS (class 1). The algorithm can be divided into two parts: multi-tree construction and multi-tree adjustment. In the multi-tree construction, SyncCast uses an approach similar to DVMA. It tries to find the latency/fairness prioritized path that satisfies both synchronization and bandwidth constraints. If there is no successful path, the algorithm relaxes the constraints (section 5.2.2). In the multi-tree adjustment, SyncCast seeks alternative paths for previous successfully assigned paths on the purpose of reducing the synchronization skews and/or the bandwidth overhead within the bounded allowance (section 5.2.2). After that, it either achieves cooperative bandwidth allocation (**BWAllocate**) at each site based on the LCF if the bandwidth constraint (section 5.2.3) is not satisfied, or delays the stream with the shortest EED before it is sent to the receiver-side buffer (section 5.3) if the synchronization constraints are violated.

Next, SyncCast decides all NDS which have been requested as AVDS (class 2). In order to reduce the bandwidth overhead at the sender sites, an option is to let these NDS receivers only request from the *serviced* sites (i.e., sites that have already received a stream as either DS or NDS). SyncCast determines all serviced sites in the AVDS topology (**DecideDSSites**) before it seeks by iteration the immediate best serviced parent site for each *unserviced* receiver (i.e. site that has not received a stream) in **SelectParentIter**.

Last, SyncCast finds possible dissemination paths for all sole-NDS (class 3). To maximize the dissemination of sole-NDS (which are usually bottlenecked at the sender sites due to the inadequate bandwidth unavailability), an option of the algorithm is to iteratively locate the closest single-hop unserviced receiver site (which is not itself a sender) to the original sender (**DecideMinOneHop**). Through the relay of each selected single-hop receiver, the algorithm then determines the dissemination paths for the rest of unserviced receivers in **SelectParentIter**.

By contrast, ViewCast only sets an upper bound on the EED and does not take synchronization into account. The algorithm simply picks a site with the least bandwidth overhead among the sender and the sites which have already received the stream. Hence, multiple streams belonging to the same sender-receiver pair can follow paths with huge EED difference and thus cause noticeable inter-stream (or inter-video) skew. Both inter-sender and inter-receiver skews, on the other hand, can only be constrained within the EED upper bound.

5.2.2 Dissemination of AVDS (class 1)

In the multi-tree construction stage, SyncCast finds the best prioritized path for each sender-receiver pair which satisfies the bandwidth constraints and the inter-sender/ inter-receiver synchronization (**SyncBW**), i.e., a successful path. If there is no successful path available, SyncCast considers three cases.

(1) If there are paths with only synchronization constraints, SyncCast selects the sync-constrained path with the least number of overloaded sites on the path (**SyncOnly**).

(2) If there are paths satisfying only bandwidth constraints, SyncCast selects the bandwidth-constrained path with the smallest inter-sender or inter-receiver synchronization skew (**BWOnly**).

(3) If there is no path satisfying either constraints, SyncCast selects the bandwidth-unconstrained path with the smallest synchronization skew (**NoConstraint**).

We call the paths assigned in the three cases above *unsuccessful* paths.

In the multi-tree adjustment stage, the function **FindAlternativePath** seeks the alternative path for the affected successfully assigned paths in order to remove/reduce the synchronization skew or bandwidth bottleneck. It takes into account the following two situations.

(1) If there is an inter-sender/inter-receiver skew violating the synchronization upper bound, SyncCast picks the successful path (and its corresponding sender and receiver) which results in the largest synchronization skew, and seeks the alternatives to the reduce the skew. This can be done via iteration until either the synchronization is constrained or no alternative path can be found.

(2) If there are sites bottlenecked at the incoming/outgoing bandwidth, SyncCast picks the successful paths which pass through the sites, and finds the alternatives by iteration until either the site is not overloaded, or no alterative path can be found.

After **FindAlternativePath** returns, if there are still unsuccessful paths violating synchronization constraints, SyncCast delays the AVDS with the shortest EED (section 5.3). If some sites are still overloaded, the algorithm applies cooperative bandwidth allocation scheme.

5.2.3 Cooperative Bandwidth Allocation

SyncCast applies the cooperative bandwidth allocation scheme to reduce the incoming/outgoing bandwidth overhead to the overloaded sites shared by multiple AVDS. We guarantee the best audio quality because users are usually more sensitive to the degradation of the audio signals. But because the bandwidth overhead of audio signals is negligible compared to the huge amounts of 3D visual information, we only discuss the video streams here.

Suppose there are L DS $(s_1, \ldots, s_L)$ that need to be relayed by an overloaded site n_i and their corresponding LCF are $(\mathrm{LCF}_{n_i}^{s_1}, \ldots, \mathrm{LCF}_{n_i}^{s_L})$, the outgoing bandwidth $B_{n_i}^{\mathrm{out}}(l)$ allocated for stream s_l at n_i can be computed as:

$$B_{n_i}^{\mathrm{out}}(l) = \frac{\mathrm{LCF}_{n_i}^{s_l}}{\sum_{m=1}^{L} \mathrm{LCF}_{n_i}^{s_m}} \times (\max B_{n_i}^{\mathrm{out}}) \qquad (8)$$

Based upon the allocated bandwidth, we then decide the frame rate of each stream by its estimated frame size which can be obtained using the linear prediction approach [6].

For the sites bottlenecked at the incoming bandwidth, we apply the same strategy by considering the LCF of the streams that are disseminating over the links between each overloaded site and its immediate parent node.

5.2.4 Dissemination of NDS (class 2 and 3)

The pseudocode of **SelectParentIter** is shown in Code 1. For each stream s in either class (from a sender site E_i^{snd}) requested by a receiver site $E_k^{rcv(i)}$, the input of the function $\mathcal{S}^{IS}$ (the set of serviced receiver sites of s) is determined by **DecideDSSites** or **DecideMinOneHop**. SyncCast searches in $\mathcal{S}^{IS}$, and picks a site n_{best}^{IS} with the smallest bandwidth overhead that satisfies the inter-stream (or inter-video) skew (**DecideBestSync**). n_{best}^{IS} then becomes the immediate parent site of $E_k^{rcv(i)}$. The new receiver site is added to the serviced set.

Code 1 Function SelectParentIter($\mathcal{S}^{IS}$)

```
1: S^rcv ← All unserviced receiver sites requesting s
2: for all E_k^rcv(i) ∈ S^rcv do
3:     n_best^IS ← DecideBestSync(S^IS, E_k^rcv(i))
4:     S^rcv ← S^rcv − E_k^rcv(i)
5:     S^IS ← S^IS + E_k^rcv(i)
6: end for
```

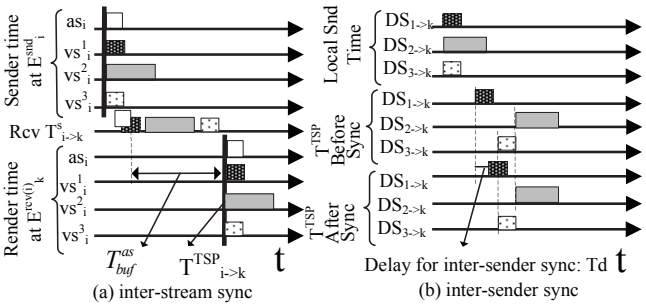

Figure 8: Receiver site buffering.

5.3 Receiver-side Buffering

Receiver-side buffering is important in SyncCast because it can (1) smooth jitters and conceal losses for audio signals, (2) synchronize the video multi-stream bundle from the same sender site before it is sent to the 3D video renderer (inter-video synchronization), (3) synchronize the audio with the corresponding video DS or multi-stream bundle sharing the same sender (audio-visual or inter-stream synchronization). We call the moment *timed synchronization point* (TSP) at a receiver site of the multi-stream bundle, which is the time that the sender's video macroframe is sent to the renderer and the synchronous audio frame to the speaker. A TSP sequence at a receiver includes all TSP for different macroframes (in time) of the same sender. Note that there can be multiple TSP sequences at a receiver site with each TSP sequence corresponding to one individual sender.

In this paper, we set each TSP based on the fact that audio signals usually require $T_{buf}^{as} = 60 - 80$ msec buffering time to smooth 98% of the audio jitters [5]. For each new video macroframe from a sender site E_i^{snd}, SyncCast searches the corresponding audio frame nearest in time (suppose the audio frame is captured at t_0 local time at E_i^{snd}). At each receiver site $E_k^{rcv(i)}$, we set the TSP for this video macroframe to $T_{i\to k}^{\mathrm{TSP}} = T_{i\to k}^{as}(t_0) + T_{buf}^{as}$ (Fig. 8(a)), where $T_{i\to k}^{as}(t_0)$ is the expected arrival time at $E_k^{rcv(i)}$ of an audio frame captured at t_0 from E_i^{snd}, and T_{buf}^{as} is set to be 80 msec in our system. Because the video DS follow the same dissemination path as the audio signal, it can arrive within T_{buf}^{as} (audio-visual synchronization). If we set $\delta_s = T_{buf}^{as}$, all video NDS from E_i^{snd} can also arrive within the inter-stream (inter-video) synchronization constraint.

To achieve the inter-sender and inter-receiver synchronization across different sender/reciever sites, all sites periodically probe for EED from every other sites. If the inter-sender or inter-receiver synchronization skew is larger than the upper bound δ_{snd} or δ_{rcv}, we delay the AVDS with the shortest EED (by a period of T_d) before it is sent to the receiver buffer, so that both inter-sender and inter-receiver synchronization can be satisfied. Fig. 8(b) shows the example of delaying $DS_{1\to k}$ (the path with the shortest EED to E_k^{rcv}) in order to reduce the inter-sender skew below δ_{snd} while guaranteeing inter-receiver synchronization. Note that we take T_d in the computation of EED in section 6.

6. EVALUATION RESULTS

We evaluate SyncCast using the one-way latency statistics in Table 2 and 3. We also compare SyncCast with ViewCast.

• Experiment Setup

We develop a multi-party 3DTI simulator for the evaluation of SyncCast. We consider both 5-site and 9-site cases, and simulate all the 12 Internet environment in Table 2 and 3. We use the link costs in the two tables to represent the latencies between the corresponding two sites in the simulator. Due to the space limit, we only consider the situation that there are roughly 50% of the sites are sender sites (i.e., 2 sender sites in the 5-site case and 4 sender sites in the 9-site case), and the senders are randomly selected in each Internet environment.

We suppose each sender site outputs 1 audio stream and 8 video streams from 8 3D cameras placed evenly (in a separation of 45-degree angle) in a 360-degree circle around a scene [20]. These cameras can capture the 3D images of the same scene from different views. From each sender site, a receiver requests 1 audio stream, 1 video DS and 2 video NDS (which are the two neighboring streams of the video DS). To simulate a real multi-site collaboration scenario, we suppose 50% of the receiver sites share the same video DS, and the other 50% of the receiver sites request different video DS. The audio streams are always requested by the receivers.

For simplicity, we represent in evaluation the incoming and outgoing bandwidth overhead as the number of 3D video streams. We assume that the bandwidth demand of audio streams is negligible compared to that of the video streams. In this paper, we set both incoming and outgoing bandwidth upper bound to be 10 video streams at each site (i.e., $\max B^{\text{in}} = 10$, $\max B^{\text{out}} = 10$). We prescribe the inter-stream skew bound δ_s be 80 msec (as discussed in section 5.3). We use an equal value for inter-sender and inter-receiver skew upper bound ($\delta_{snd} = \delta_{rcv}$). To simulate different application demand, the upper bound varies from 100 msec to 300 msec (in a separation of 100 msec).

To compare SyncCast with ViewCast, we also constrain the ViewCast's inter-sender and inter-receiver synchronization skews within the same upper bound as SyncCast, by assigning the value of δ_{snd} (or δ_{rcv}) to the EED upper bound in ViewCast.

• Simulation Results

Victims. Fig. 9(a) shows the number of victim sites in ViewCast. The victims are incurred due to the fact that (1) ViewCast does not take into account service class differentiation, so the limited bandwidth resources can be used to transmit less visually-important NDS; and that (2) a small EED upper bound can reduce AVDS path availability, so the number of victim sites increases as the EED upper bound decreases. By contrast, there are no victim sites introduced by SyncCast because of the service class differentiation, cooperative bandwidth allocation and AVDS path delaying policy to satisfy the inter-sender/inter-receiver synchronization constraints. Fig. 9(b) shows the number of delayed AVDS paths. As the inter-sender/inter-receiver synchronization upper bound decreases, more AVDS paths with short EED will create an unbounded synchronization skew and thus need to be delayed. Fig. 10(a)-(b) show that the number of victim streams are consistently lower than ViewCast due

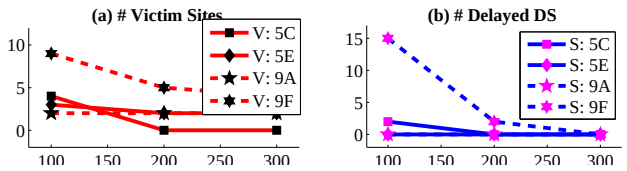

Figure 9: SyncCast/ViewCast. X-axis is δ_{snd} (δ_{rcv}).

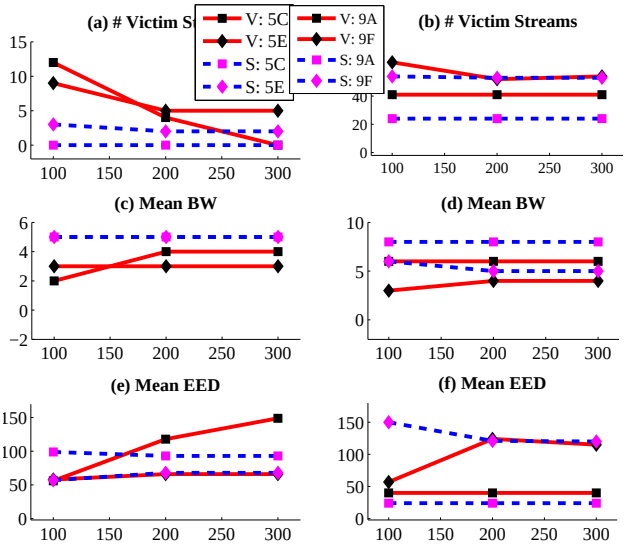

Figure 10: SyncCast/ViewCast. X-axis is δ_{snd} (δ_{rcv}).

to the combined impact of the service class differentiation, routing fairness and bandwidth preservation policies.

Bandwidth overhead. The average incoming and outgoing bandwidth overhead should be equal because we assume there is no broken link in this paper. Fig. 10(c)-(d) demonstrate that SyncCast boasts a high bandwidth utilization than ViewCast and thus a smaller number of victim streams/sites.

Average EED of AVDS paths. Fig. 10(e)-(f) show the average system EED of all AVDS paths of the two algorithms. When $\delta_{snd} = \delta_{rcv} = 200$ or 300 msec, SyncCast's results are comparable to or smaller than ViewCast's data. This shows the effectiveness of SyncCast algorithm based on DVMA. When $\delta_{snd} = \delta_{rcv} = 100$ msec, due to huge amounts of unsuccessful transmissions of AVDS (victim sites) in ViewCast, the remaining successful AVDS paths with short EED can lead to an average smaller than SyncCast's results. As mentioned in Section 3, the minimization of the number of victim sites is placed at the top priority in the 3DTI system in order to guarantee AVDS's audio-visual contributions.

Inter-stream skew. We compare the inter-stream skews in Fig. 11(a)-(b). We show that SyncCast can consistently achieve the inter-stream synchronization within the preset 80 msec constraint by dropping streams which will cause unbounded skew. On the other hand, ViewCast exhibits randomness of the inter-stream skews which can exceed the constraint, because it only sets the EED upper bound.

Inter-sender and inter-receiver skews. Fig. 11(c)-(f) show the maximum inter-sender and inter-receiver skews. They show that both ViewCast and SyncCast can successfully limit all the skews within the prescribed upper bound. Note that because the two protocols are using different ap-

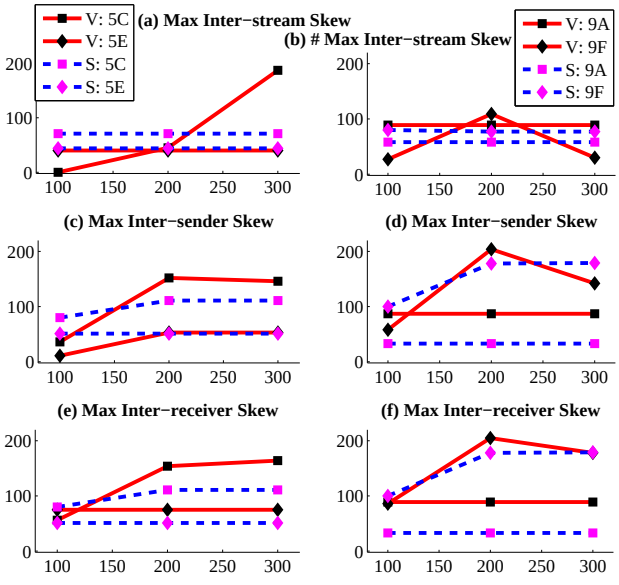

Figure 11: SyncCast/ViewCast. X-axis is δ_{snd} (δ_{rcv}).

proaches to achieve the synchronization, the resulting skews of SyncCast do not have to be smaller than those of ViewCast. In addition, the synchronization upper bound of ViewCast depends on its preset EED upper bound. Hence, more victim sites/streams (due to a lack of paths with bounded EED) can be introduced under the circumstance of a tight timing synchronization constraint.

7. CONCLUSION

In conclusion, compared to ViewCast, SyncCast can successfully reduce the number of victim sites and victim streams, minimize the average system EED of all DS paths, achieve the bandwidth and synchronization constraints. Our new disseminations scheme is not limited to the 3DTI application. It can be extended to any delay-sensitive online gaming systems when each avatar can output more than one correlated media streams, and the timing synchronization is critical.

We would like to acknowledge all constructive comments from the reviewers. This research study is supported by NSF CNS 0720702 and 0834480, by UPCRC grant from Intel and Microsoft, and by Grainger Grant.

8. REFERENCES

[1] P. Agarwal, R. Rivas, W. Wu, and K. Nahrstedt. Bundle of streams: Concept and evaluation in distributed interactive multimedia environments. In *Proc. of IEEE Int'l Symposium on Multimedia*, 2010.

[2] A. Bhattacharya and Z. Yang. Dvbmn-l: delay variation bounded multicast network with multiple paths. In *Proc. of IEEE Int'l Conference on Multimedia and Expo*, pages 762–765, 2009.

[3] E. Brosh, A. Levin, and Y. Shavitt. Approximation and heuristic algorithms for minimum-delay application-layer multicast trees. *IEEE/ACM Transaction on Networking*, 15(2):473–484, 2007.

[4] R. Hassin and A. Tamir. On the minimum diameter spanning tree problem. *Information Processing Letters*, 53:109–111, 1997.

[5] Z. Huang, B. Sat, and B. W. Wah. Automated learning of play-out scheduling algorithms for improving the perceptual conversational quality in multi-party VoIP. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 493–496, July 2008.

[6] Z. Huang, W. Wu, K. Nahrstedt, A. Arefin, and R. Rivas. Tsync: A new synchronization framework for multi-site 3d tele-immersion. In *Proc. ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, June 2010.

[7] S. Khuller, B. Raghavachari, P. State, and N. Young. Balancing minimum spanning trees and shortest-path trees. *Algorithmica*, 14:305–321, 1995.

[8] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *American Mathematical Society*, 7(1):48–50, 1956.

[9] F. A. Kuipers. An overview of constraint-based path selection algorithms for qos routing. *IEEE Communications Magazine*, pages 50–56, 2002.

[10] K. Liang and R. Zimmermann. Cross-tree adjustment for spatialized audio streaming over networked virtual environments. In *Proc. ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 73–78, 2009.

[11] T. Little. A framework for synchronous delivery of time-dependent multimedia data. *Multimedia Systems*, 1(2):87–94, 1993.

[12] H. Liu and M. Zarki. An adaptive delay and synchronization control scheme for Wi-Fi based audio/video conferencing. *Springer Wireless Networks*, 12(4):511–522, July 2006.

[13] G. N. Rouskas and I. Baldine. Multicast routing with end-to-end delay and delay variation constraints. *IEEE Journal on Selected Areas in Communications*, 15(3):346–356, Apr. 1997.

[14] B. Sat, Z. Huang, and B. W. Wah. The design of a multi-party VoIP conferencing system over the Internet. In *Proc. IEEE Int'l Symposium on Multimedia*, pages 3–10, Taichung, Taiwan, Dec. 2007.

[15] B. Sat and B. W. Wah. Playout scheduling and loss-concealments in VoIP for optimizing conversational voice communication quality. In *Proceedings of ACM Multimedia*, pages 137–146, Sept. 2007.

[16] S. Y. Shi, J. S. Turner, and M. Waldvogel. Dimensioning server access bandwidth and multicast routing in overlay networks. In *Proc. of ACM International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 83–92, 2001.

[17] R. Steinmetz. Human perception of jitter and media synchronation. *IEEE Journal on Selected Areas in Communications*, 14(1):61–72, 1996.

[18] Z. Wang and J. Crowcroft. Quality of service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14:1228–1234, 1996.

[19] Z. Yang. Multi-stream synchronization for 3d tele-immersive and collaborative environment. In *Proc. of ICST Conference on Immersive Telecommunications*, 2009.

[20] Z. Yang, W. Wu, K. Nahrstedt, G. Kurillo, and R. Bajcsy. Enabling multi-party 3d tele-immersive environments with viewcast. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 6(2), 2010.

[21] Z. Yang, B. Yu, K. Nahrstedt, and R. Bajscy. A multi-stream adaptation framework for bandwidth management in 3D tele-immersion. In *Proc. ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2006.

[22] K. Yasumoto and K. Nahrstedt. Ravitas: Realistic voice chat framework for cooperative virtual spaces. In *Proc. of IEEE 2005 International Conference on Multimedia and Expo*, 2005.

[23] R. Zimmermann and K. Liang. Spatialized audio streaming for networked virtual environments. In *Proc. of ACM Int'l Conference on Multimedia*, pages 299–308, 2008.