Uncompressed High Definition Visual Sharing System towards Interactive Networked Collaboration

Jinyong Jo [*]
HPcN Project Div.
Korea Institute of Science and Technology Information
jiny92@kisti.re.kr

JongKwon Chae
Dept. Info. & Comms.
Gwangju Institute of Science and Technology
jkchae@nm.gist.ac.kr

Minsun Lee
e-Science Div.
Korea Institute of Science and Technology Information
mleeoh@kisti.re.kr

JongWon Kim
Dept. Info. & Comms.
Gwangju Institute of Science and Technology
jongwon@nm.gist.ac.kr

OkHwan Byeon
e-Science Div.
Korea Institute of Science and Technology Information
ohbyeon@kisti.re.kr

ABSTRACT

Recent growth in available network bandwidth envisions widespread use of broadband applications such as uncompressed Internet HDTV, a cutting-edge application technology for networked collaboration. In this paper, we introduce a cost-effective uncompressed HD visual sharing system and discuss its software architecture in depth. We also illustrate our experiences with the visual sharing system for interactive collaboration. Finally, experimental results will show how well it compensates for packet jitter to gear up media synchronization.

Keywords

Uncompressed high definition video, immersive media, interactive conferencing, tele-presence, and network-based collaboration.

1. INTRODUCTION

Grid computing enables us to do faster calculations and process larger data sets by utilizing high performance computing resources connected by broadband networks. With the explosive growth of network bandwidth and computing performance, network engineers have turned a focus to the search for an application utilizing broadband network infrastructure. The demands are getting combined with visualization technologies such as Access Grid [1], IPTV [2], uncompressed Internet HDTV (high definition TV) [3],[4],[5] and networked virtual reality.

Uncompressed Internet HDTV is a cutting-edge system requiring advanced networking and application technologies. As a tradeoff of consuming huge (but will become abundant and cheaper) bandwidth, it provides a number of merits over other compression-based visual sharing applications[1]. Media compression makes it difficult to develop a real-time high-quality video transport system because of processing delay caused by codec and its processing complexity. Also, the compressed media streams are likely to be weak for transport errors due to dependency among video frames. On the other hand, uncompressed HD video is strong to the error propagation and produces no additional delay for the codec processing.

In this paper, we present the extensive design and implementation of a cost-effective uncompressed Internet HDTV system by continuing our previous effort [6]. The usability of proposed realization is also illustrated by showing our experiences in various scientific and technical areas of collaboration. Contributions of proposed realization include a low-cost media interface, *pseudo* circuit emulation and a separate audio session, dual-port transmission with software packet-striping, and efficient audio-playout support. These contributions are meaningful in that the proposed system is highly cost-effective to make itself prevail for the advanced visual sharing of future collaboration environments.

The rest of this paper is as follows. We first look around a brief overview of the proposed system and its networking features, and introduce our experiences in Section 2. The system design and implementation issues are followed in Section 3. In Section 4, performance evaluation results are provided for both short- and long-haul IP networks. After discussing related works in Section 5, we conclude this paper in Section 6.

2. OVERVIEW AND EXPERIENCES

Our goal in developing the application technology is to

[*] A corresponding author.

[1] With the term 'visual sharing', we are referring the natural and immersive exchange of all visual (video as well as image/graphics) and associated audible media streams among collaboration participants. However, still in this paper, it is quite synonymous with high-quality video conferencing.

IMMERSCOM 2007, October 10-12, Verona, Italy
Copyright © 2007 978-963-9799-06-6
DOI 10.4108/ICST.IMMERSCOM2007.2149

provide users with a high-definition visual sharing environment at low implementation cost. To this end, we take advantage of an open-source licensed software, UltraGrid [5] and improve it to be cost-effective.

We develop a low-cost high-definition media interface, which grabs or plays high quality video and audio, and take deep consideration on delivering audio in a flexible way. High definition 1080i video with uncompressed format can be delivered over a dual-port GbE (Gigabit Ethernet) card (i.e., 2×1 Gbps). The dual-port transmission is another economical feature of the proposed technology. We also introduce *pseudo* circuit emulation and build up a two-phase synchronization to provide tuned playout. Finally, media playout with graphic and sound card makes the proposed system be more competitive in a sense that the common peripheral devices can substitute costly media interface.

The application technology requires high-performance systems and advanced networking technique to manage real time traffic in bulk. For instance, the system has to process more than 18,000 packets per second when we apply jumbo frames (i.e., 9,000 bytes packets). The number of packets to be handled increases in 600% in case of 1,500 bytes packets. The small-size packets lead to frequent hardware interrupts and memory-copy operations, and they steeply raise system load to deteriorate transmission throughput. Like this, jumbo frame is essential in the proposed system.

Burst traffic from the proposed system has an *on*-and-*off* duration within a frame time (1/fps) and in-bulk jumbo frames have tendency to overwhelm normal IP traffic [6]. Within burst *on* period, transmission rate would reach line capacity and then inter-packet departure space becomes minimum 8 bytes due to Linux IPG (inter-packet gap) configuration. To put a packet on wire, we have to attach overhead to the packet (18 bytes at data link layer, 8-byte preamble and minimum 8-byte IPG at physical layer). It indicates normal IP packets are hardly multiplexed with jumbo frames which are in burst *on* period. It leads our expectation to minimized packet jitter in the period. Increase in delay jitter happens right after the *off* duration because normal traffic may take network resources at the moment.

There are three networking methods we used to apply in: best-effort in layer 3, port-based VLAN (virtual LAN) [7] in layer 2 and user-controlled lightpath (UCLP) [8],[9] in physical layer. VLAN is used to divide a network into pieces for traffic management and security purpose. Port-based VLAN is easy to configure but needs agreements on VLAN IDs in advance. UCLP allows network users to provision and reconfigure lightpaths in physical networks within a single domain or crossing multiple domains. With UCLP, users can dynamically create or tear down their own logical IP network.

We introduce our experiences with the proposed system. The experiences will briefly show what areas our work contributes to. Fig. 1 is a snapshot of dancing performance, which is multicasted from Asia to Europe via North America. The goal of this event was cultural exchange which needs vivid sense of presence. We first established lightpaths from Asia to North America and North America to Europe, and bound them with a unique VLAN ID. A rendezvous point was picked up to disseminate multicast packets.

Also, in medical area requiring ultra-precision images and low latency, the application technology can carry out important roles. From engineering point of view, medical training has many advantages over other application areas. First, we can define a service domain to local network. It simplifies network configurations to make easy interconnection between a trainer (e.g., surgeon) and trainees (e.g., doctors-in-training or residents). Another attraction is that medical equipment usually have a high-definition microscopic camera. Endoscope or laparoscope camera gives good examples. Fig. 2 shows laparoscopic liver surgery in an operating room. Two cameras grabbed macro-/micro-scopic view of the operating room and internal organs. Two views were multiplexed and unicasted to a conference room. We applied 8-bit 1080i video (995 Mbps except audio) and downsized it to have 980 Mbps including audio and used an endpoint fragmentation and reassembly. Modern GbE cards are capable of fragmenting a large packet in pieces and reassembling the segmented frames. Endpoint fragmentation must be limited to use in local network in case of no jumbo frame support since it produces much reassemble failure in long-haul networks.

Figure 1: Experiences: Dancing performance.

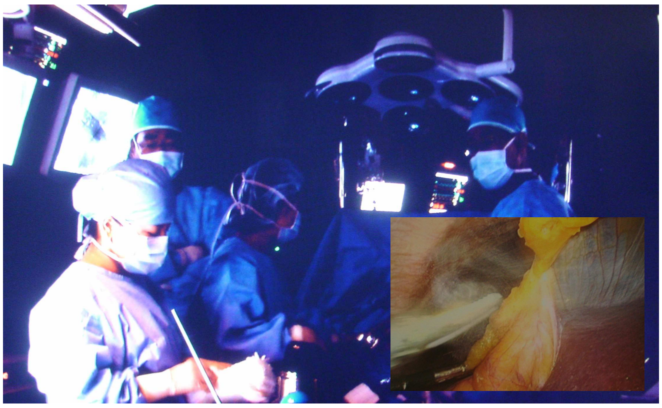

Figure 2: Experiences: Laparoscope surgery.

3. DESIGN AND IMPLEMENTATIONS

3.1 Software Architecture

The goal of UltraGrid is to provide high-quality interactive video conferences along with general purpose HD distribution and visualization [10]. Though UltraGrid is capable of transporting either progressive (720p) or interlaced

(1080i) HD video, it has originally focused on 720p format in real use. HDstation or Centaurus cards from Digital Video Systems (DVS) [11] capture or playout uncompressed video. It is also possible to display 720p video on a window using a graphic card. Adaptation of various video codecs allowed UltraGrid to have more flexible features. A video frame from UltraGrid is encapsulated in RTP packets and delivered through a single-port GbE interface.

UltraGrid has potentials to support two types of RTP payload formats: Circuit emulation [12] and native packetization [13]. Circuit emulation packetizes and transports SMPTE292M signal to get compatible with other industrial products using HD-SDI interfaces. Native packetization can pick up various video formats by means of packetizing active video samples from SMPTE signal. UltraGrid hired the native packetization for uncompressed video. It supports IPv4, IPv6 and 1-to-many IP multicast. In addition, UltraGrid does not consider high-quality audio other than exploiting external toolkits like RAT (Robust Audio Tool).

We have improved the software architecture of UltraGrid to make it hold cost-effective system features (see Fig. 3). It provides a media interface for 8- and 10-bit high-definition 1080i video as well as 6-channel audio (24-bit 48KHz PCM). We defined a RTP payload header, which separates an audio stream from networked high definition video, to present channel information. Our architecture employed an approach to use a dual-port GbE card in order to make a traffic load-sharing. 10-bit video (data rates up to 1.32 Gbps or 1.485 Gbps) can be smoothly delivered with the dual-port transmission. In addition, we support multi-channel audio playout by using ALSA (Advanced Linux Sound Architecture) [14] driver.

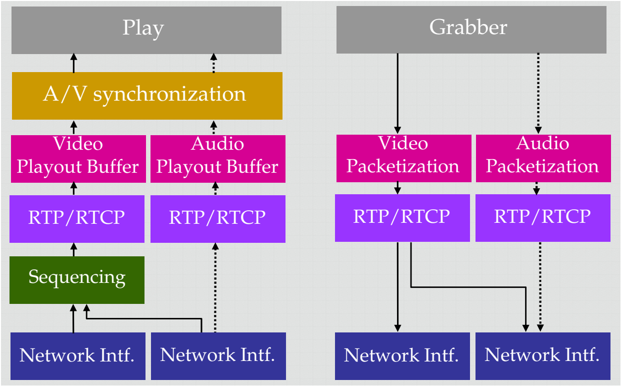

Figure 3: Software architecture of proposed system.

Media Capture and Playout

We picked OEM HS from AJA video systems [15] to capture and playout high-definition video and 6-channel audio. We also employed common media interfaces (e.g., graphic and sound cards) for playout to reduce implementation cost. Since a separate audio stream issues intra- and inter-media synchronization, we, by the time of capturing, adjusted a frame rate of audio to that of video (i.e., 29.97/59.94, 30.0/60.0 and 25.0/50.0 fps) to ease the synchronization. It also facilitates reconstitution of SMPTE292M signal. We define it as *pseudo* circuit emulation.

Existing uncompressed HD systems [3],[4],[5] have transported either circuit-emulated SMPTE292M signal or active samples of video. None of them has dealt with separate audio session. No need for inter-media synchronization (i.e., lip synchronization) is the profound advantage of circuit-emulated SMPTE292M. It however requires relatively high network bandwidth and makes down flexibility for visual sharing purposes. In addition, it is impossible to make a separate audio session from circuit-emulated SMPTE292M.

With the separation of the sessions, we can raise the flexibility in application use and save network bandwidth though we have to deal with increased complexity to make inter-media synchronization. In addition, native-packetized audio and video streams make it easy to directly play each medium with off-the-shelf peripheral devices such as graphic and sound card.

Media Transport and Reconstruction

Exploiting 10 Gbps is still a costly solution to approach broadband networks. One of the simple yet cost-effective methods is to use multiple network cards. Proposed architecture takes advantage of dual-port transmission, which binds two Gigabit network cards to support transmission rate over 1 Gbps. The dual-port transmission enables us to provide 10-bit uncompressed high-definition video services over bandlimited networks.

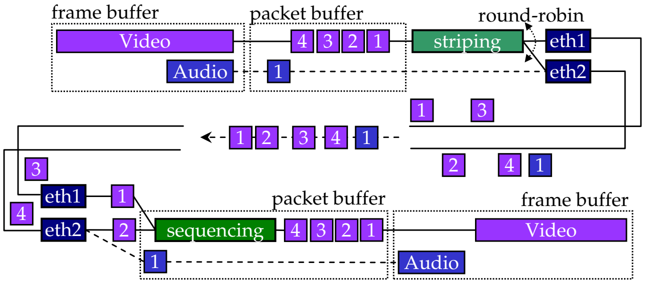

Figure 4: Software packet-striping.

We implemented a software packet-striping like Fig. 4 to achieve the dual-port transmission. Similar ideas were introduced in [16][2]. The packet-striping treats two network cards as a single virtual interface. Ideally, it takes about 23 ms (i.e., fall under the burst *on* period) at wire speed to transmit a single 10-bit video frame. After packetization, video packets are pushed over two GbE cards (i.e., eth1 and eth2) with round-robin fashion. Thus, traffic load gets fairly distributed on two network cards. To avoid mutual interference between video and audio traffic, an audio frame is sent right after the end of a video frame.

Dual-port transmission is hard to ensure sequential packet arrivals at receiving parts. Out-of-ordered packet arrival is very common in IP networks though its occurring frequency is very low. Mistaking out-of-ordered arrivals in the proposed architecture would be sometimes originated from low timing accuracy (e.g., 1∼10 ms) of Linux operating systems.

[2] Turning into a link aggregation protocol (IEEE 802.3ad) can be an alternative approach. Judging from our experiences, it however proved to have compatibility issues between host systems and network switches, to make a large amount of out-of-ordered packets, and to increase implementation complexity.

Non-blocking *select()* function often commits the mistakes. Note that inter-arrival space of video packets is normally limited to a couple of microseconds. Thus, we have to reconstruct original packet sequence. The proposed architecture looks into two RTP packets yielding concurrent events and rearranges them to have original packet order (i.e., sequencing in Fig. 3 and Fig. 4). A packet being late is inserted in place unless it breaks expected presentation time.

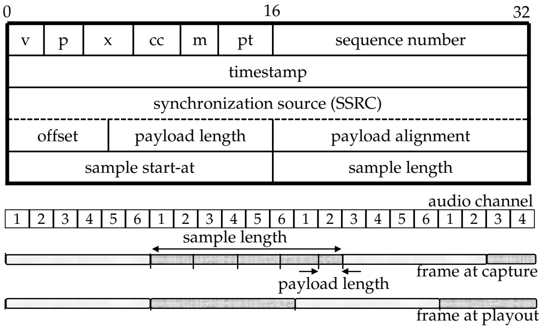

Figure 5: Audio RTP and payload header.

We also designed a RTP payload header for audio (see Fig. 5) to make synchronization easily and to cope with channel loss robustly. Note that payload header for uncompressed video format is defined in [13]. When we grab an audio frame with *pseudo* circuit emulation, the number of samples per audio frame will be variable (e.g., the number of samples equals to the truncation of 48KHz/29.97fps and a surplus.). The `offset` field in Fig. 5 reflects the surplus. `payload alignment` indicates a correct place of the payload to be inserted. A number of packets comprise an audio frame. `start sample-at` implies the start channel number of the audio frame and it prevents channel swapping. By the time of a channel loss, we can maintain a channel to speaker mapping based on the field.

Jitter Compensation and Media Synchronization Challenge in maintaining multiple sessions is to minutely keep presentation time between media (i.e., inter-media synchronization) and to observe a presentation interval within a medium (i.e., intra-media synchronization). Matters are more difficult in high-definition television since even a minimal presentation error results in image flickering. Factors acting on the presentation error include packet jitter, losses and low timing accuracy of non-realtime operating systems.

Network congestion incurs packet jitter to cause the presentation error. We delay presentation time of a frame to absorb jitter. A frame is held up to $1/f_s + 3 \cdot j_t$ seconds from the arrival time of the first packet comprising the frame. f_s is the average frame rate of video. Delay jitter, j_t, is calculated by an exponential moving average. Packets arriving after the deadline will be regarded as loss to maintain real-time features of the system.

We introduce two-phase media synchronization. First phase is issued in frame buffers (i.e., frame-level synchronization) and second phase is done in a playout buffer (i.e., media-level synchronization). The reason why we used two-phase synchronization is that timing accuracy of Linux is unsatisfying to make strict synchronization. Since imprecise RTP and NTP (network time protocol) timestamp have only a couple of millisecond precision, they ought to be used for the frame-level synchronization. RDTSC (read time-stamp counter) timestamp grants more accurate timing for media-level synchronization. The RDTSC timestamp guarantees at least a micro-second precision.

Inter-media synchronization at frame-level compensates for asynchronization caused by delay jitter, frame loss, and so forth. A medium framed in advance keeps waiting until its correspondent completes frame composition. The waiting time is no more than 45 μs over 1 GbE since one audio frame produces 5 packets sized 9,000 bytes each. We then look into expected presentation time of each medium to allow a limit of synchronization skew (e.g., 33 ms). If the skew is going beyond the limit, we just discard the old one waiting for inter-media synchronization. Discarding a frame is the simplest way to minimize processing delay and system load.

Media frames already synchronized are conveyed on a playout buffer. By the time the frames are copying into the buffer, they are given both the same RDTSC timestamp and their memory addresses. A data structure wraps the timestamp and the addresses to be used for media-level lip synchronization. Since we can think of the data structure as a single medium, we only have to ensure a precise intra-media synchronization. Software interrupts issued by a RDTSC clock rearrange exact presentation time (i.e., intra-media synchronization) of the medium. Initially, two frames (i.e., $2/fps$ initial buffering time) are logged into the playout buffer to prevent frame underflows and to fully guarantee the intra-media synchronization.

3.2 Hardware and System Tuning

We integrated hardware platforms on top of commercially off-the-shelf powerful systems. They comprised Dual-Xeon processors or a Dual-Core Opteron processor on PCI-X mother boards, and ran Redhat 9 (Linux 2.4.25) or Mandrake 10.1 (Linux 2.6.11.3) operating system.

Table 1: System and interface tuning.

```
# max amount of option memory buffers
echo 524288 > /proc/sys/net/core/optmem_max

# default & max values for the rx/tx window.
echo 262114 > /proc/sys/net/core/wmem_default
echo 262114 > /proc/sys/net/core/rmem_default
echo 128388607 > /proc/sys/net/core/rmem_max
echo 128388607 > /proc/sys/net/core/wmem_max

# max number of incoming packets to be queued
echo 6000 > /proc/sys/net/core/netdev_max_backlog

# set large MTU & avoid local congestion
ifconfig eth<x> mtu 9,180
ifconfig eth<x> txqueuelen 9,000
```

OEM HS from AJA video systems provides a HD-SDI (serial digital interface) to capture and playout media frames. As alternatives, high-end graphic and sound cards enable multimedia frames to be played out on PC monitors and speakers. Either a 10 Gigabit (Chelsio N210 [17]) or a dual-port GbE (2×1 Gbps) card delivers the captured media frames to remote destinations.

We enforced OS kernel tuning and others (e.g., MTU size, Interrupt coalescence and so forth) to maximize system performance. OS kernel tuning adjusts read/write buffer-size to prevent packet overflows. Make sure that system tuning is an influential factor to the performance (e.g., MTU of small size makes system load increased and small queue length brings excess packet loss. Both deteriorate playout quality.). A tuning script is shown in Table 1.

Additional tuning to the dual-port or 10 GbE card is required to optimize Interrupt Coalescence [18] or increase Maximum Memory Read Byte Count (MMRBC). Interrupt Coalescence rearranges packet-arrival spacing and MMRBC controls PCI-X burst transfer size. These are not directly coupled with improving transmission throughput but alleviating high system load. Mitigating load raises the extensibility of the proposed system (e.g., we can take benefits such as implementing a system which simultaneously handles more than two streams.). We set the values based on the results of our previous works in [6].

4. PERFORMANCE EVALUATION

We conducted a number of experiments over short-/long-haul Internet to observe correct operation of the proposed system and to investigate the effect of video to audio traffic. No network provision was made up in advance. Verifying them over best-effort IP network is significant since many factors (loss, delay, and jitter) may exert harmful influence on the system performance.

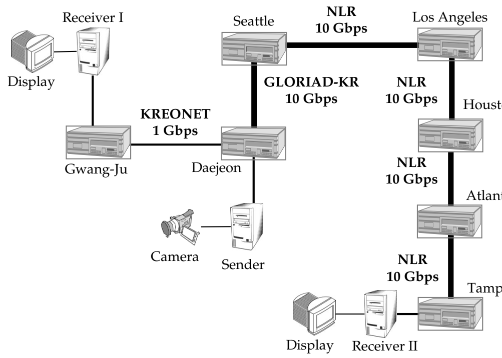

Figure 6: Testbed configuration.

Fig. 6 illustrates a network-testbed for the experiments. For tests over short-haul Internet, we configured a 1 Gbps dedicated link between two nodes (sender and receiver `I`). Background traffic was at 2 Mbps and round-trip time (RTT) was less than 4 ms. Long-haul tests were made up during the Super Computing 2006 conference. Traffic traversed 10 Gbps GLORIAD (GLObal RIng network for Advanced application Development) and NLR (National Lambda Rail) to reach the destination (receiver `II`). There was 430 Mbps background traffic and RTT was 231 ms. We conducted both the tests over best-effort IP networks (no lightpath provisioning) and the links produced no packet loss. Uncompressed 1080i video and 24-bit audio traffic with 9,000-byte packet size including header were crossing the networks for 10 minutes. Sending rate was tuned to 980 Mbps.

We plotted frame rate of video in Fig. 7 to observe how well the proposed architecture carries out media synchronization. In the short-haul Internet, 69.3% of video frames were captured at every 1/29.97 second and 88.5% were played out at the rate (90.8% of frames at capture and 97.9% at playout time fell in ±50 μs from 1/29.97 second interval.). In the long-haul Internet, 72.8% and 98.8% of frames kept the precise rate while capturing and playing the frames respectively (both are within 99.5% if we allow ±50 μs.).

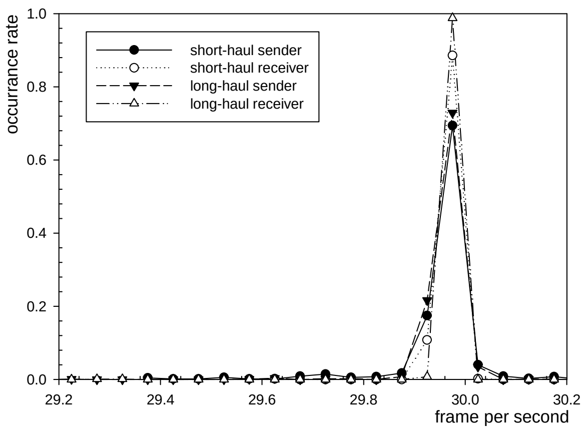

Figure 7: Frame per second.

The result indicates the proposed architecture well gears up intra-media synchronization. It shows that the receiving parts absorbed errors in frame rate occurred at capture time. Non-blocking *select()* and low timing accuracy in Linux bring on inexact frame rate (i.e., low framing accuracy) at capture time. In the proposed architecture, a process waits for packet arrivals for 20 μs prior to sending a frame. The 20 μs blocking causes the imprecision. One thing remarkable is that the framing accuracy at playout time becomes lower as backbone bandwidth goes narrower. It is because jitter of audio shown in Fig. 8 triggers additional delay for inter-media synchronization.

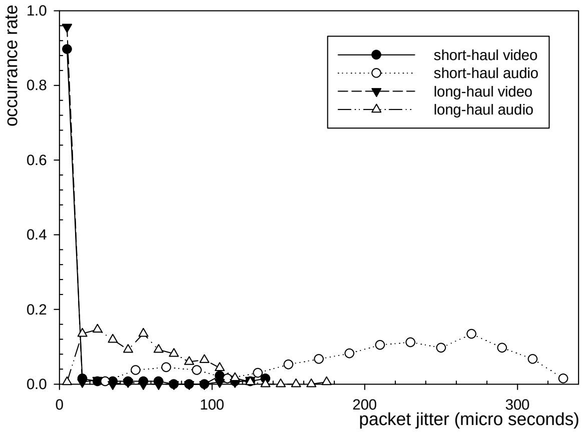

Figure 8: Packet jitter.

Over 90% of video packets show less than 6 μs jitter without regard to the backbone bandwidth and the distance. Uncompressed video traffic used to be extremely burst (e.g., 14,000 9,000-byte packets per second) and do not allow other traffics to be multiplexed in. The behavior keeps jitter of video packets low. In contrary to the video traffic, audio packets are vulnerable to the burstness. We can see 97% of audio packets has less than 100 μs jitter in broadband (i.e., 10 Gbps) long-haul network, whereas 88% of the packets shows jitter greater than 100 μs in narrowband (i.e., 1 Gbps) short-haul network. The result implies that media quality is rather dependent on backbone bandwidth than distance.

We also observe that video traffic with uncompressed format has tendency to keep low jitter due to its burst nature. Audio traffic however is more vulnerable to competing video traffic as backbone bandwidth becomes narrower. Judging from the result in Fig. 7, the proposed architecture successfully makes up for the jitter to gear intra-/inter-media synchronization.

5. RELATED WORKS

We briefly describe existing works to implement uncompressed HDTV systems. Note that there are a number of other existing variants but we just select only a few approaches which are very well-known. Software architectures can group the existing works: iHD1500 [3], UltraGrid [5], i-Visto [4].

Research Channel has developed iHD1500, which supports 10-bit 1080i video. Since the software transmits circuit-emulated SMPTE292M signal over IP networks, it needs at least 1.485 Gbps network bandwidth for one direction. The software runs on Microsoft Windows^{TM}. iHD1500 shows around 4-frame end-to-end delay (roughly, 133 ms) to process audio and video frames. As a cost-effective feature, frames in iHD1500 are fairly transported over dual-port network interface.

USC/ISI has researched and released an open-source package for uncompressed HDTV, UltraGrid. Though they use 10-bit 720p video in real events, internal transport architecture can accommodate 1080i video in either circuit-emulated SMPTE292M or native packetization. UltraGrid has brought RFC 3497 and RFC 4175, which are standard transmission formats of uncompressed HDTV. The standard formats promote transport flexibility of UltraGrid.

NTT has released a commercially available uncompressed HDTV system called i-Visto (internet-Visual studio system). With i-Visto, multimedia frames can be transmitted over SONET/SDH as well as Gigabit Ethernet and up-to 3 media streams are concurrently processed by the system. Trunking multiple network interfaces in i-Visto increases transport throughput to 1.6 Gbps.

6. CONCLUSIONS AND FUTURE WORKS

We introduced a cost-effective high definition visual sharing system and discussed our experiences with it. Our experiments taken over best-effort IP networks demonstrate that audio traffic is subject to be vulnerable to burst video traffic, and the behavior brings on increased packet jitter over narrowband IP network. We also observed that the proposed system makes well up for the packet jitter to ensure media synchronization.

Cost-effectiveness of the system has to be further granted to make it prevail in R&E areas. In addition, we need to look into the effect of network congestion to the system performance, and to investigate how the burst traffic makes influences on normal Internet traffic.

7. REFERENCES

[1] L. Childers, T. Disz, R. Olson, M. Papka, R. Stevens, and T. Udeshi, "Access Grid: Immersive group-to-group collaborative visualization," in *Proc. 4th International Immersive Projection Technology Workshop*, 2000.

[2] B. Alfonsi, "I wan my IPTV: Internet protocol television predicted a 'Winner'," *IEEE Distributed Systems Online*, vol. 6, no. 2, 2005.

[3] J. DeRoest and M. Wellings, "Digital well and HD projects," CESNET Conference, 2006.

[4] K. Mochida, T. Kawano, T. Ogura, and K. Harada, "The i-Visto gateway XG - uncompressed HDTV multiple transmission technology for 10 Gbit/s networks," Tech. Rep., NTT, 2005.

[5] USC/ISI, "UltraGrid: A high definition collaboratory," http://ultragrid.east.isi.edu/, 2006.

[6] J. Jo, W. Seok, M. Lee, and O. Byeon, "Optimizing system performance for uncompressed HDTV over 10-gigabit Ethernet," *Journal of Korea Information Processing Society(Korean)*, vol. 13-c, no. 5, 2006.

[7] Timothy M. O'Neil, "Network-based quality of services," White paper from Polycom Video Communications, 2002.

[8] D. Linh Truong, O. Cherkaoui, H. Elbiaze, N. Rico, and M. Aboulhamid, "A policy-based approach for user controlled lightpath provisioning," in *Proc. IEEE/IFIP Network Operations and Management Symposium 2004*, Apr. 2004.

[9] J. Jo, W. Hong, S. Lee, D. Kim, J. Kim, and O. Byeon, "Interactive 3D HD video transport for e-science collaboration over UCLP-enabled GLORIAD lightpath," *Journal of Future Generation Computer Systems*, vol. 22, no. 8, pp. 884–891, Oct. 2006.

[10] L. Gharai, T. Lehman, A. Saurin, and C. Perkins, "Experiences with high definition interactive video conferencing," in *Proc. of IEEE International Conference on Multimedia & Expo (ICME)*, July 2006.

[11] DVS Digital Video Systems, "Centaurus HDTV capture cards," http://www.dvs.de/english/products.

[12] L. Gharai, C. Perkins, G. Goncher, and A. Mankin, "RTP payload format for society of Motion Picture and Television Engineers (SMPTE) 292M video," IETF RFC 3497, Mar. 2003.

[13] L. Gharai and C. Perkins, "RTP payload format for uncompressed video," IETF RFC 4175, Sept. 2005.

[14] ALSA Project, "Advanced linux sound architecture," http://www.alsa-project.org/.

[15] AJA video systems, "OEM HS HD-SDI interface," http://www.aja.com/.

[16] A. Hari, G. Varghese, and G. Parulkar, "An architecture for packet-striping protocols," *ACM Trans. on Computer Systems*, vol. 17, no. 4, pp. 249–287, Nov. 1999.

[17] Chelsio Communications, "High-performance 10 Gb Ethernet adapters," http://www.chelsio.com/.

[18] Intel, "Interrupt moderation using intel gigabit ethernet controllers," http://www.intel.com/design/network/, 2003.