

# Network Delay Estimation in Reverse Genlock Synchronized Display Walls

Jochen Miroll, Yongtao Shuai and Prof. Dr.-Ing Thorsten Herfet, Saarland University and Intel Visual Computing Institute, Saarbrücken, Germany. E-Mail: {miroll,shuai,herfet}@nt.uni-saarland.de

## Abstract

A tiled display wall is a combined screen composed of multiple individual displays. Without compromising display resolution, a scalable solution for building tiled display walls is proposed. Herein, each display node is solely equipped with an Internet protocol (IP) connection. As display nodes are independently free running, synchronization is required at the display refresh level and regarding the content. In this paper, we focus on the aspect of network forwarding delay estimation, which ultimately determines the accuracy of display phase synchronization of a tiled display wall following the Reverse Genlock paradigm as proposed by the authors earlier. We show that in both switched and shared medium networks, forwarding delay may be estimated with microsecond accuracy.



<http://www.intel-vc.i.uni-saarland.de/en/projects/hybrid-rendering.html>

## 1. Introduction

The demand for high resolution visual display has increased rapidly with developments in science and technology, particularly with respect to display manufacturing processes, switching circuit integration density, computer processing power and video coding. While Full-HD (1080p) video resolution is common for home television, the first publicly available Ultra-HD (4K resolution) television broadcast has been launched in January 2013<sup>1</sup>. Furthermore, multi-view, as e.g. stereoscopy, has become popular in high resolution visual reproduction of moving images. With increasing demands, the possibility of combining multiple individual (potentially multi-view capable) high resolution displays to create a joint canvas for video display at even further increased resolution is becoming increasingly interesting. Modern LED-backlit displays feature narrow bezel sizes in the order of few millimeters, enabling tight attachment of individual displays next to each other. However, with increasing number of displays of a so-called tiled display walls, scalability becomes an issue.

Commodity personal computer graphics devices may support multiple displays by means of multi-head, in which multiple display devices are connected to a single source. Daisy-chaining of video signals and consecutive partitioning of the signal at the display device is another frequently encountered technique. Above named hardware based solutions for tiled display walls exhibit poor scalability in terms of resolution and display area. Alternatively, software-based solutions using personal computer hardware [1, 2] have been proposed to overcome limitations of dedicated hardware solutions. In software, the challenge is the synchronization of content display across multiple independent display nodes, whereas a *node* is the combination of a personal computer and a connected display device (such a projector or liquid-crystal display). As a television broadcast compatible solution which may further reduce costs and effort of building tiled display walls, we have proposed a solution involving purely IP interconnected “Smart Internet Displays” [3]. IP provides great flexibility of connecting displays and content sources (wired or wireless) as it removes the need for additional cables such as dedicated synchronization wires. By using compressed video signals and network streaming, scalability is improved and mobility is introduced. Broadcast compatible hardware further introduces a major benefit: Television broadcast is fully synchronized from source to sink by means of program clock reference timing information being transmitted with the digital audio-visual data and usage of homogeneous oscillators to generate base frequencies for recording, broadcast and playback. Synchronization accuracy as compared to that achievable with PC hardware is thus increased. Nevertheless a tiled display wall should be autonomous from any existing television broadcast station, while non-TV content such as computer generated moving images (e.g. Ray-Tracing) shall be displayed synchronously across the wall of individual display nodes. We have therefore proposed an IP-based “reverse Genlock” solution for tiled display walls [3], in which the tiled display wall acts as the synchronization source (clock master) for the generation of visual content.

<sup>1</sup> <http://www.eutelsat.com/products/broadcast-ultra-hd.html>

## 2. Problem Statement

Display walls for digital signage can be found frequently at airports and shopping malls. Many of the solutions seem to be daisy-chained setups, in which a single video signal (e.g. 1080p resolution) is passed through the displays and each of them displays a certain part of the video signal. However, the video signal is limited in bandwidth as determined by the video connection standard (e.g. HDMI or DisplayPort). Without compromising display resolution, multi-head solutions multiply the available video signal bandwidth by introducing additional video output ports. However, when scalability towards a large number of displays is targeted, the use of multiple video ports (such as those named above, WiDi or 802.11ac) may become infeasible or quite expensive. With respect to synchronization, any daisy-chained or multi-head solution is driven by a single base frequency and synchronization is immanent. For individually running, solely IP interconnected display nodes, this is not the case.

Synchronization in visual systems can be described by three indispensable levels with the following terms. *Genlock (Generator Locking)*: is a technique where a video sink synchronizes to the clock of a single generator. A typical example from Digital Video Broadcasting (DVB) is that the DVB receiver will decode and playback the audio-visual content at the exact same rate of the content being generated. *Swaplock*: ensures that the content being displayed across multiple applications is synchronized. *Framelock*: is used to synchronize the refresh of a visual frame on any two display devices. For a tiled display wall composed of individual display nodes to display images synchronously, any two of the devices actually displaying images (LCD monitors, TVs, projectors, etc.) have to be synchronized to compensate for differences in visual refresh rate and phase.

### 2.1. Frequency Synchronization

One possible straightforward approach applicable to IP based synchronization follows the master-slave paradigm, in which any one of the displays is selected as the clock master display (CMD) and the rest of the displays (slaves)

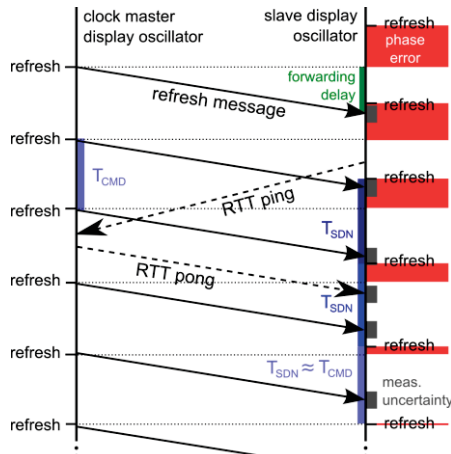


Figure 1: IP based master-slave display frequency and phase synchronization.

synchronize accordingly. If the display nodes are indeed capable of matching their visual output frequency and phase to pre-defined values adaptively, a combined screen can be composed of these display nodes. In the following, an IP network-based reverse Genlock mechanism to synchronize display refresh rates and times across composite displays (Framelock), and video content (Swaplock) is described as previously proposed [3]. The term “reverse” refers to the fact that the roles of the content generator and the content sink are inverted with respect to synchronization: the sink provides a clock signal to the source(s), which adapt content generation speed. Fig. 1 illustrates synchronization by example: a CMD periodically transmits UDP/IP messages at each of its own refresh time instants. Based on those herein called display clock reference (DCR) IP packets each slave display adjusts its refresh rate to match that of the CMD. The CMD’s exact visual refresh frequency can be estimated by evaluating arrival timestamps said packets as they are assumed to arrive at equi-distant time intervals [6]. On a non-realtime scheduling operating system and in combination with best-effort IP transport, noise is added to arrival timestamps, introducing the need for proper filtering [3].

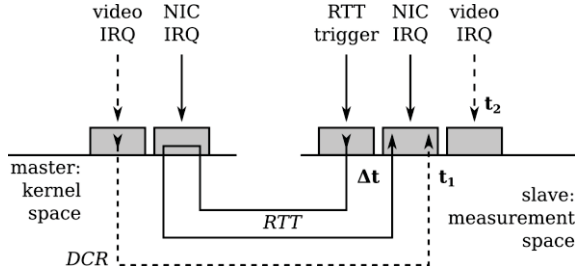
### 2.2. Phase Estimation

While our previous work covers frequency synchronization, the focus of this work is on the estimation of packet forwarding delay in the IP network. Even with frequency identity of display refresh, the phase offset between a slave display node and the CMD remains unknown. To estimate it, each slave display (and content sources as well, if need be) may periodically measure the network round trip time (RTT) to the CMD to evaluate the forwarding delay of the display refresh UDP/IP messages. Only with a perfectly accurate forwarding delay estimate and frequency identity, slaves may achieve phase identity with the CMD at all times. It is important to note that, subject to some maximally achievable accuracy, a tiled display wall has to be in sync in both frequency and phase beyond the point where a human observer cannot tell a difference. We consider the evaluation of subjective tests as future work.

Note further that internal processing of heterogeneous sets of displays may differ. In this case, phase equalization is still possible but requires extrinsic information.

### 2.3. Content Synchronization

With proper synchronization of any two display nodes in display frequency and phase, Swaplock is achieved by provision of timeline information in DCR packets. By comparison of DCR timeline information with video frame timestamps during playback, synchronous video playback is achieved across the display wall. In the same manner, content sources synchronize their content generation speed with the composite screen. The sum of all rendering, buffering and processing times will result in a delay between source(s) and sink(s) that is to be determined for any discrete setup. Automatic detection and optimization of this delay sum is beyond the scope of this work.



**Figure 2:** Master-slave synchronization with packet forwarding delay estimation by RTT measurement. Timestamps  $t_1$  and  $t_2$  are used for frequency synchronization,  $\Delta t$  additionally for phase offset compensation. All timestamps are taken within one operating system layer (e.g. kernel space).

### 3. Refresh Rate Adaptation

The main problem of synchronization of two independently running oscillators (e.g. driving the video outputs of display nodes) is frequency synchronization. Naturally, any two oscillators differ in their exact frequency, resulting in phase drift. Thus, the goal is to remove the frequency difference. For this, we have previously evaluated mechanisms to modify display frequencies by two fundamentally different approaches. In case the base frequency provided by an oscillator is tunable, i.e. a voltage controlled crystal oscillator (VCXO) is implemented and its voltage is software-controllable, frequency synchronization is possible in software in case the tunable range of the respective two oscillators is overlapping. If such fine granular frequency tuning is unavailable, modification of the uncompressed video signal transmitted from graphics adaptor to display device via e.g. VGA or HDMI is the only other possibility for frequency adaptation [2, 3]. Yet it has turned out that modification of neither active nor inactive signal components [7] is tolerated by a wide variety of available displays [3]. For this reason, broadcast television hardware with software-controllable VCXO circuitry is used herein.

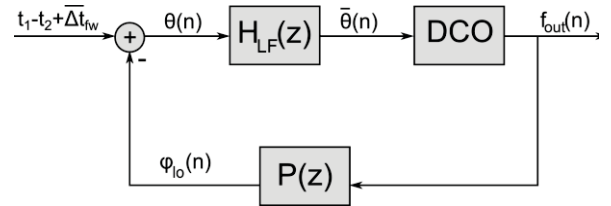
### 4. Forwarding Delay

In the system described above, the arrival times of DCR packets are used to determine both frequency and phase. However, the phase obtained is subject to a phase shift composed of the forwarding delay of the DCR packets as well as operating system scheduling.

Fig. 2 depicts timestamps involved in the IP based synchronization. The CMD issues a DCR message per video interrupt (IRQ). The forwarding delay of DCR messages on the IP network is unknown when arrival timestamps are taken in the network interface card (NIC) IRQ. Hence, a slave display node issues RTT measurement packets that are responded by the master immediately (echoed). For frequency synchronization the timestamps  $t_1$  and  $t_2$  are used, while for phase equalization  $\Delta t$  is compensated additionally. In order to compensate for different scheduling latencies at different operating system levels (cf. kernel-space vs. userspace), timestamps are taken within one

operating system level at each node. Note that although the CMD may take timestamps in the kernelspace, this is not mandatory for slave devices. Furthermore, a single clock source is used for time measurement at each node. This can be the NIC's clock or the system clock, as we solely consider time differences in the order of milliseconds and the whole system is free of any absolute time.

In fig. 4 the software phase locked loop (PLL) is depicted with the average forwarding delay estimate  $\Delta t_{fw}$  introduced into the algorithm. For details on the PLL design, the reader is referred to our previous work [3]. Consequently, the accuracy of estimating the network forwarding delay heavily affects the performance of the phase compensation. It is important to note that on a non-idle network the forwarding delay may be time-dependent. As it determines display phase, it needs to be known at all times. Similar to TCP [4], our implementation uses a first-order recursive filter, i.e. an exponentially weighted moving average, for RTT estimation. Improved results can be obtained by a Kalman filter [5] design. Filtering of RTT by said means is valid as there is a stable mean RTT during times of quasi-static traffic conditions (as assumed herein). The forwarding delay, as derived from the RTT, is fed pre-filtered into our software PLL in order to maintain loop stability.



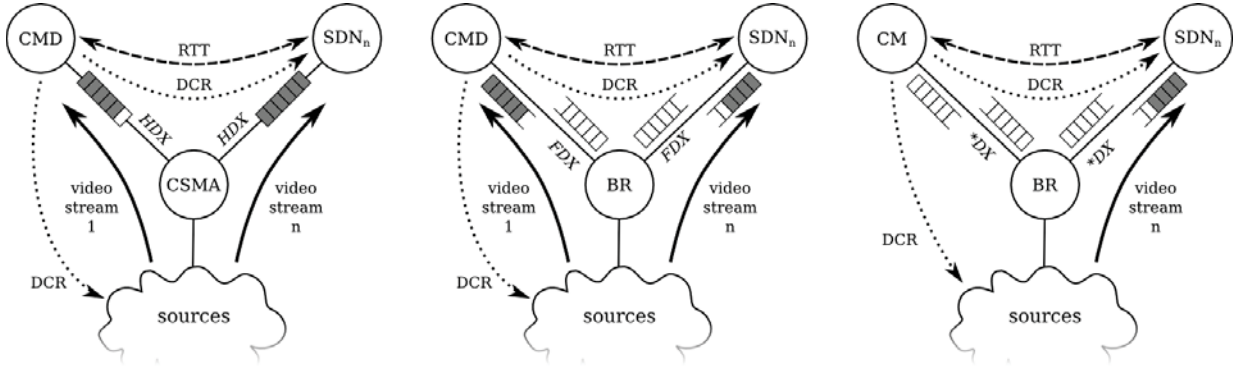
**Figure 3:** Phase-Locked-Loop

#### 4.1. Round-Trip-Time Measurement

In order to compute the network forwarding delay from a known RTT, further knowledge about network topology is necessary. In one case, whenever during RTT packet traversal of the network the forwarding delays in both directions, i.e. from slave to master and vice versa, are identical, the RTT is symmetric. With different networks and traffic conditions, it may be asymmetric [9, 10]. Note that the underlying network technology and the traffic load, i.e. buffer filling levels of potentially independent inbound and outbound queues, may be determining factors for symmetry.

#### 4.2. Multiple Access Networks

In telecommunications, a connection between two end points for transmission of digital data is either Simplex (SX), Half-Duplex (HDX) or Full-Duplex (FDX). While SX describes unidirectional communication (cf. broadcast), HDX and FDX are bidirectional. Former provides a single channel shared between at least two end points for communication in both directions, while FDX provides two separate channels, one per direction, between



**Figure 4:** Symmetric and asymmetric IP networking scenarios, by which display nodes and content sources may be connected. Areas shaded in gray symbolize occupied queue slots. *Left:* Half-Duplex (HDX) CSMA network; all packets traverse the same queues for each link. *Middle:* Full-Duplex (FDX) bridged (BR) network; queuing is dependent on traffic flow direction. *Right:* Clock master not being a display, thus not receiving video streaming traffic.

exactly two end points.

Legacy Ethernet hardware (hubs) are employing carrier sense multiple access (CSMA) with collision detection (CD) and connected network interface cards (NIC) had been HDX devices connected via a shared bus. Wireless LAN (cf. 802.11a/b/g/n/ac) is operating in HDX mode since CSMA with collision avoidance (CSMA/CA) is employed on the shared wireless medium. Fast Ethernet (cf. 1000BASE-TX) on the other hand is operating in FDX mode, in which a bi-directional orthogonal point to point connection is established by a network switch to each of the attached network terminals. In both above named examples, orthogonality of transmission for multiple users is achieved by time division multiple access (TDMA). However, medium access collision domains are link-individual in switched networks due to multiport bridging [12]. Fig. 4 depicts the different scenarios within the scope of this work: a) CSMA (left); b) bridged (BR) FDX (middle); and c) bridged HDX/FDX with asymmetric streaming (right). In fig. 4 it is further shown how RTT is measured by e.g. a slave display node  $n$  in order to derive the network forwarding delay of the periodic DCR messages.

For single hop CSMA HDX systems such as a local wireless LAN we assume symmetry of network forwarding delay. In absence of network load figures, knowledge about network topology and potential interference, we believe that no exact recipe can be given for determining forwarding delay. Thus, we distinguish two cases: Firstly, in case of plain CSMA networks we compute network forwarding delay due to symmetry as

$$\overline{\Delta t}_{fw} = RTT / 2 \quad (1)$$

In switched networks on the other hand, an important role is played by the traffic load on each bridged FDX or HDX link.

### 4.3. Network load

In the scope of this work, video streaming towards a set of display nodes is considered. As depicted in fig. 4 (center), RTT measurement via a multi port bridge can not necessarily be assumed symmetric in terms of bi-directional forwarding delays. Delay from the CMD to a slave may be

different from vice-versa due to differences in queue occupancy. However, for phase equalization, the DCR one way delay needs to be determined exactly.

Herein, the primary focus is on homogenous sets of displays, i.e. they are provided with video at identical spatio-temporal resolution. Thus it is assumed that video streaming traffic towards each display node is identical in throughput on average. Consequently, scenario b) is assumed symmetric and network forwarding delay is computed as given in equation (1).

In scenario c), the clock master is not a display node itself. As shown in fig. 4 (right), RTT measurement is clearly asymmetric as there is an imbalance in video streaming traffic on involved FDX links. Specifically, RTT is asymmetric during periods in which the slave display nodes receive video traffic. When there is a period of time over which no video streaming traffic is present on the network (e.g. in an initialization phase), a minimum RTT should be determined on the otherwise idle network and the forwarding delay is thus found as

$$\overline{\Delta t}_{fw} = \min(RTT) / 2 + [RTT - \min(RTT)] \quad (2)$$

## 5. Results

We use the network simulator ns-3, which is an open and highly validated environment for network simulation, to determine whether the above assumptions with respect to RTT symmetry are true. As elaborated above, synchronization accuracy in display refresh of any two displays is determined by RTT estimation accuracy. In software however, we may not know whether the estimated RTT yields the correct DCR forwarding delay (when computed correctly from RTT as described above), or not (cf. RTT asymmetry). When the result is incorrect, however, in our case we would be able to determine this by observing the display output in terms of display phase. Consequently, we provide results as externally measured.

### 5.1. Network Simulation (ns-3)

We set up two scenarios of RTT estimation in ns-3. Firstly, based on the topology shown in fig. 4 (left), the ns-3 simulation is a star network including one CMD node, nine

slave display nodes, a video server node. The link between the nodes is a CSMA (half-duplex) channel with a data rate of 100 Mbit/s. This scenario (A) is symmetric as described above. Secondly, the CSMA channel is replaced by a switched<sup>2</sup> (multi-port) full-duplex network at the same speed (scenario B). In both scenarios, the CMD transmits DCR packets at the visual refresh frequency (here: 60 Hz). Each slave display node requests a UDP echo from the CMD every 1/10 seconds to estimate the RTT to the CMD. The network is otherwise idle. Starting at the half of the total simulated time, the video server streams UDP traffic to each of the display nodes individually with average bit rate of 6 Mbit/s each, corresponding to state-of-the-art high-definition video. The simulation lasts 60 seconds. Scenario B is created asymmetric as the video server does not stream video to the clock master node.

The following table provides results of the forwarding delay estimation using eq. (1) in the symmetric case and eq. (2) in the asymmetric case. Simulation enables determining both forwarding delay and RTT exactly in both cases. We obtained on average about 6% error in forwarding delay estimation in the symmetric case, and 2.1% in the asymmetric case.

Delay est.	N1	N2	N3	N4	N5	N6	N7	N8	N9
Sym. (1)	1.10	1.048	1.056	1.046	1.027	1.054	1.057	1.065	1.078
Asym. (2)	0.995	1.022	0.967	1.021	0.999	1.046	1.027	0.981	1.011

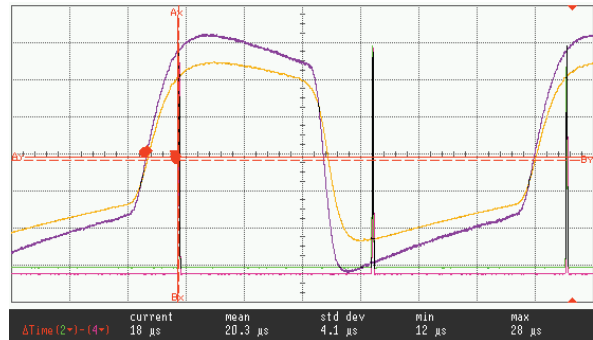
This shows that proposed RTT estimation is reliable in the case of larger numbers of displays and with realistic video streaming traffic conditions.

## 5.2. Measurements

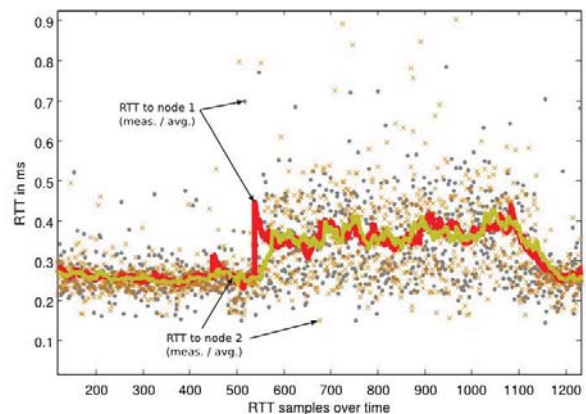
We measure the accuracy of forwarding delay estimation and consequently, phase equalization externally by using DVI/HDMI vertical blanking pulse detectors inserted into the video cables. Furthermore, we have put photo-diodes in front of the displays and feed the displays with periodically alternating black/white video sequences at the frequency identical to the display refresh rate. In fig. 5 an oscilloscope trace is shown in which the sync pulses and the black/white transitions as captured by the photo-diodes are shown for two displays receiving artificially generated streaming video network traffic at link saturation. The clock master is external and not receiving video traffic, resulting in an asymmetry. Forwarding delay is thus computed as given in eq. (2) and compensated as shown in fig. 3. Note that in figure 5, six narrowly overlapping spikes are displayed (over two refresh periods at 60 Hz extracted from two video cables each). LCD transition curves are taken from identical displays but are of different amplitude due to practical reasons. The achieved phase error at an average of 20  $\mu$ s is almost three orders of magnitude smaller than the refresh period.

<sup>2</sup> <https://codereview.appspot.com/5615049>

In a video<sup>3</sup> we show how frequency and phase are syn-



**Figure 5:** Synchronization accuracy comparing the display output of two displays. Spikes are derived from HDMI signals, curves are photo-diode voltages detected from alternating black/white test sequences as displayed on connected LCDs. *X-Axis:* 5ms/div; *Bottom:* Synchronization accuracy.



**Figure 6:** Network round trip time measurements from two display nodes to a clock master. At  $x \approx 500$  both respective network links are saturated with background traffic, network is idle before and after  $x \approx 1050$ . Temporal averaging reveals the RTT is stable during both periods.

chronized at two nodes, one of which estimates forwarding delay according to eq. (1) although there is a forwarding delay asymmetry, cf. fig. 4 (right). Consequently, a phase error is introduced at this node. Respective other node correctly uses eq. (2).

## 6. Display as a Service

Synchronization in frequency and phase as presented herein shall be described in a broader context. Löffler et al. [11] describe a software framework implementing a virtual frame buffer that resides in the network. This buffer may be of arbitrary pixel dimensions and serves as an abstraction for the concept of the Display as a Service (DaaS). For example, tiled display walls composed of either homogeneous or heterogeneous individual display devices can be represented therein. Applications may write pixel data into the virtual frame buffer, while pixels are displayed by the DaaS on an arbitrary set of displays located on a 2D plane. Within DaaS, the reverse Genlock paradigm is applicable in order to synchronize both ends: pixel

<sup>3</sup> <http://www.youtube.com/watch?v=wZcFuYID8Yw>

producing applications (potentially distributed over the network) and the individual display devices.

## 7. Concluding Remarks

In this paper we have shown how IP based synchronization of independently running displays can be achieved in frequency and phase. We have identified different prevailing network scenarios resulting in different approaches to network forwarding delay estimation. The main question answered herein is whether forwarding delay in a non-idle network can be estimated sufficiently accurate for increasingly large numbers of displays. In conclusion, the solution is feasible with an initialization phase being necessary in most cases. Its duration depends on the number of displays, network traffic and topology conditions. Subjective evaluation of synchronization accuracy is considered future work.

## Acknowledgements

The work presented in this paper has been partially financed by the Intel Visual Computing Institute at Saarland University. The content is under sole responsibility of the above named paper authors.

## References

- [1] J. Allard, V. Gouranton, G. Lamarque, E. Melin, and B. Raffin, "SoftGenLock: active stereo and genlock for PC cluster," in Proceedings of the workshop on Virtual environments, 2003, p. 260.
- [2] M. Waschbüsch, D. Cotting, M. Duller, and M. Gross, "WinSGL: software genlocking for cost-effective display synchronization under Microsoft Windows," Proceedings of the Sixth Eurographics Symposium on Parallel Graphics and Visualization, 2006.
- [3] J. Miroll, A. Löffler, J. Metzger, P. Slusallek, and T. Herfet, "Reverse genlock for synchronous tiled display walls with Smart Internet Displays," IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin), pp. 236–240, 2012.
- [4] V. Jacobson and M. J. Karels, "Congestion avoidance and control," ACM SIGCOMM Computer Communication Review, vol. 18, no. 4, pp. 314–329, 1988.
- [5] K. Jacobsson, H. Hjalmarsson, et al. "Round-Trip time estimation in communication networks using adaptive Kalman filtering," Reglermöte, 2004.
- [6] Y. Stein, R. Shashoua, R. Insler, and M. Anavi, "Time Division Multiplexing over IP (TDMoIP)." RFC 5087 (Informational), Dec. 2007.
- [7] Digital Display Working Group Promoters, "Digital Visual Interface – DVI, Revision 1.0," 1999.
- [8] Nirnimesh, P. Harish, and P. J. Narayanan, "Garuda: a scalable tiled display wall using commodity PCs," IEEE transactions on visualization and computer graphics, vol. 13, no. 5, pp. 864–77, 2007.
- [9] D. Wei, S. H. Low, J. Bunn, H. D. Choe, J. C. Doyle, H. Newman, S. Ravot, S. Singh, F. Paganini, G. Buhmaster, L. Cottrell, and O. Martin, "FAST TCP: from theory to experiments," IEEE Network, vol. 19, no. 1, pp. 4–11, Jan. 2005.
- [10] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, "Measurement considerations for assessing unidirectional latencies," Internetworking: Research and Experience, vol. 4, pp. 121–132, 1993.
- [11] A. Löffler, L. Pica, H. Hoffmann, and P. Slusallek, "Synchronous Networked Displays for VR Applications: Display as a Service (DaaS)," Joint Virtual Reality Conference of ICAT - EGVE - EuroVR (JVRC), 2012.
- [12] A. Tanenbaum, "Computer Networks (4th ed.)", Prentice Hall Professional Technical Reference, 2002