Challenges for Selecting Optimal Coordinators in Peer-to-Peer-based Massively Multi-user Virtual Environments

Richard Süselbeck, Florian Heger, Gregor Schiele and Christian Becker
Universität Mannheim
Mannheim, Germany
{ richard.sueselbeck | florian.heger | gregor.schiele | christian.becker }@uni-mannheim.de

***Abstract*—Peer-to-Peer-based Massively Multi-user Virtual Environments (P2P-MMVEs) often use coordinator peers to enhance the system's capabilities. The success of this approach depends on selecting suitable peers as coordinators. However, existing approaches either do not select coordinators based on their suitability or simply assume that techniques for such a selection exist. In this paper we argue that these techniques are not currently available for P2P-MMVEs. We believe that developing such techniques is one of the few remaining challenges for the real-world feasibility of P2P-MMVEs. As a first step towards this goal, we identify and discuss the challenges associated with this research in detail.**

I. Introduction

Peer-to-Peer-based Massively Multiuser Virtual Environments (P2P-MMVE) often use so-called coordinators, for example to improve scalability [1]–[10], security [11] or consistency [12] [13]. A coordinator is a special peer, that is selected from the set of all peers at runtime to take over specific tasks for the system. As an example, to improve scalability, a coordinator can coordinate the exchange of messages between peers. Similarly, to improve security, a coordinator can act as a trusted device that monitors the behaviour of other peers and tries to detect attackers.

To achieve good system performance, it is important to select the right peers as coordinators. If a peer with low bandwidth is selected to coordinate the message exchange, it may become a bottleneck and slow down the system. If a monitoring coordinator is an attacker itself, the security of the system may be compromised. Thus, such peers should not be selected for these tasks. The selection algorithm must take this into account and compare the suitability of different peers with respect to the specific requirements of the target system.

Despite the fact that coordinators are commonly used in P2P-MMVEs, very few systems actually select peers as coordinators based on their suitability. Often coordinators are selected based on simple techniques such as their overlay ID (e.g., [2]) or their position in the virtual world (e.g., [14]), which are unrelated to their suitability. Some approaches acknowledge the specific requirements of their coordinator peers, but simply assume the existence of techniques to determine whether a peer fulfils them (e.g. [10] [4]).

We thus argue that coordinator selection in peer-to-peer systems is not a solved problem. In fact, as selecting suitable coordinators is a necessity for many P2P-MMVEs, we consider this to be one of the few remaining steps towards the real-world feasibility of P2P-MMVEs.

We believe that the key reason why selecting suitable coordinators remains challenging, is that in it can very difficult to actually determine the values of many of the selection parameters in a P2P system. Issues that need to be dealt with include highly fluctuating values, measurement overhead and the reliability of measurements, all in a decentralized environment.

In this paper we analyze the coordinator selection problem in P2P-MMVEs in more detail. We focus on the problem of determining correct values for the selection parameters and identify and discuss unsolved issues and open questions. The goal of our paper is to encourage the research community to address these unresolved issues and to keep the discussion on coordinator election going.

The paper is structured as follows: First we present our system model in Section II. Then we discuss the key parameter properties in Section III. Subsequently we discuss potential parameters in the context of these properties in Section IV. In Section V give an example of the challenges of coordinator selection in the context of our own P2P-MMVE system. Finally we present related work in Section VI and give an outlook on future work in Section VII.

II. System Model

For this discussion we assume that the peers of the P2P-MMVE run on end user computers. These systems are heterogeneous with regard to a variety of resources and properties, such as CPU performance and bandwidth. The P2P-MMVE software uses a significant amount of these resource, e.g., for 3D graphics and AI routines. The computers also run an a priori unknown number of other applications in parallel with the MMVE software. The MMVE must share the system's resources with these applications.

The systems are connected to the Internet using consumer-level Internet access. The bandwidth capacity provided by this access is also heterogeneous, ranging from low-bandwidth connections with asymmetric upstream/downstream to to high bandwidth connections with symmetric upstream/downstream. Access is often established via a residential gateway (e.g., a

978-1-4577-0638-7 /11/$26.00 ©2011 IEEE

wireless router) with network address translation (NAT) and/or firewall functionality. In addition, access may be shared with other applications on the system or even other systems in the user's home network.

For this paper we assume that the coordinator selection algorithm uses a utility function to determine the fitness of a peer to serve as coordinator. All peers in the system are possible candidates. When a peer has been selected as coordinator, it continues to perform the basic functions of a peer in addition to the coordinator functions.

III. Key Properties of Utility Parameters

We have identified three key properties of utility parameters which we believe make determining a peer's utility challenging. These properties are Cost, Fluctuation and Reliability. In the following we will discuss each of them in detail.

A. Cost

Determining the current value of a utility parameter is associated with overhead. This means that determining the utility of a peer uses the resources of the system and comes with a cost to its performance. Due to the nature of peer-to-peer system, i.e. due to to churn, coordinator selection is a regular occurrence. It is therefore important to consider the cost of determining each parameter.

There are several factors which must be considered. First, we need to consider which resources are being used when determining a parameter's value. Each measurement puts load on one or more resources of the system. For example, determining the maximum bandwidth of the Internet connection requires sending data, thus putting load on the connection.

After considering which resources are used in a measurement we must determine how much load the measurement puts on these resources. Checking the current free space on the hard disk, for example puts load on both CPU and hard disk. This load is small however. As such this parameter can be determined with little impact on system performance. On the other hand, many algorithms for determining the maximum bandwidth of a connection require sending as much data as possible over the Internet connection for a period of time. During this time, the measurement can use the entire bandwidth available to the system. This means that the peer may not be able to perform its functions while this measurement takes place. In short, the cost of some measurements can be prohibitively high.

In addition, the cost of certain measurements increases with the quality of the result. For example, when determining the maximum bandwidth, sending data across the link for longer periods of time produces a more precise result. In turn, the bandwidth cost of the measurement increases.

Finally, we must consider not just the cost of the measurement on the peer which initiates it, but also the cost on all other peers which are involved in the measurement. While some measurements, such as determining current CPU load only involve a single peer, others involve one or more additional peers in the system. For example, the cost of determining the network delay to another peer is low. This cost is not just incurred on the peer initiating the measurement however, but also on the peer the delay is measured to. The fact that several peers can be involved in a measurement means that we must also consider scalability as a factor. For example, to accurately determine the network delay parameter of a peer, we would have to measure its delay to not just one, but all other peers in the system. In an MMVE with a massive number of peers, this does not scale.

B. Fluctuation

All parameters fluctuate over time, i.e. their value does not remain the same. As a consequence each value has to be measured repeatedly. There are three aspects that have to be considered when looking at the fluctuation of a parameter: the value's variance, the frequency of the value's changes and the magnitude of the value's changes.

Variance indicates how much the value of a parameter can deviate from the mean. Consider the available bandwidth of a system. This value can vary significantly, depending on the amount of bandwidth used by all applications and systems utilizing the peer's Internet access. On the other hand, the number of a system's CPU cores has no variance at all, as the number of cores is fixed. If a value's variance is low, this means that the value does not have to be measured often, as it will remain close to any previously measured value. Low (or no) variance may even make additional measurements unnecessary. Depending on the frequency and magnitude of changes, high variance however indicates that the value may need to be measured repeatedly.

The frequency of changes indicates how often a parameter's value changes. Frequent changes can necessitate regular re-measurements of a value, particularly if the parameter's variance is high. The current CPU load on a system changes frequently, depending on the applications which are currently running and the task they are performing. The maximum bandwidth of a system only changes however, when the user changes his type of Internet access, which is a rare occurrence.

Finally, the magnitude of changes indicates how large the individual changes to the parameter's value can be. Some parameters may change frequently, but each individual change of the value tends to be small. An example of this is the available hard disk space. Changes are frequent, but usually small in relation to the overall disk space. For other parameters each change can be large, such as for available bandwidth, where stopping a single application can significantly change the value of the parameter. Parameters with large magnitude of change require more frequent re-measurements of their value. This is the case even when the parameter's variance and frequency of changes is low, as a change with large magnitude can significantly impact the overall peer utility. Note that low variance excludes a large magnitude of changes, though a parameter with high variance may have both small and large magnitudes of changes.

C. Reliability

Not every measurement of a parameter's value necessarily returns a correct result. While some values, such as remaining hard disk space can be determined exactly, measuring others relies on algorithms that return imprecise result. When looking at an unreliable value, we cannot be sure that is accurately represents the state of the parameter.

A good example for this is the maximum bandwidth. There are a variety of algorithms for determining bandwidth, but the reliability of their output depends on various factors, such as whether other systems are using the Internet connection. Some of these factors are outside the control of the peer, e.g., a bottleneck along a network route. It is therefore possible to get different results each time the bandwidth is measured.

The degree of reliability depends on the individual factors that cause it and must be determined individually for each parameter. In the worst case, a parameter may be so unreliable as to be unsuitable for determining the utility of a peer. It is possible however, to mitigate some unreliability factors at the expense of additional overhead. For example, repeatedly measuring network and averaging the results can reduce the effects of jitter on the reliability of the measurement, but places additional load on the network connection.

Note that an unreliable value should not be mistaken for a value that is no longer correct due to fluctuations. An incorrect value can be updated through another measurement. The next measurement of an unreliable parameter is just as unreliable as the first one.

D. Discussion

The only parameters whose value can always be determined efficiently and reliably are those which have low cost, low fluctuation and high reliability. As very few parameters fulfil these requirements, we must find approaches that allow us to strike a compromise between these three properties. To do so we must consider the following questions:

First, how do we measure the parameter? Some types of measurements are clearly prohibitive for cost reasons alone, but there may be alternatives which strike an acceptable balance between cost, fluctuation and reliability. For example, actively measuring the maximum bandwidth of a peer can use all available bandwidth on a peer for a while. This can make it impossible for the peer to function during this time. As an alternative, passive measurement could be used, which monitors only existing network traffic and deduces the maximum bandwidth based on the data gathered. In this case reliability is sacrificed in order to reduce the cost of the measurement.

Second, when and how often do we measure a parameter? The frequency with which we measure a parameter has significant impact on overall cost. Even a high cost measurement may be acceptable, if the measurement has to be performed only at very long intervals. The exact time when a measurement is performed can also be relevant. Timing a measurement to occur during a phase with low load can mitigate some of its cost. Even otherwise prohibitive measurements may be acceptable, if they only have to be performed during start-up.

Finally, if and only if we cannot find an sufficiently efficient and reliable way to measure a parameter, we must consider if we can create a suitable utility function which does not include this parameter. In some cases the cost may be so high or the reliability so low as to be prohibitive.

IV. Utility Parameters

We will now look at specific parameters and discuss them with regard to the properties we have just discussed. Based on existing work (see Section VI) and our own analysis we have identified the following list of potential parameters: CPU performance, system memory, long-term storage, stability, reputation, bandwidth, delay and connectivity restrictions.

We classify these parameters in three different categories: a) parameters based on the peer's system (e.g., CPU performance and storage space), b) parameters based on the user's behaviour (e.g., reputation and uptime) and c) parameters based on the peer's network connection (e.g., delay and bandwidth). We will discuss each of these categories and their parameters in the following.

A. System Resources

Parameters in this category are based on the local hardware resources of the peer's system. The parameter are CPU performance, memory space and long-term storage.

1) CPU: The CPU performance of a peer is often a key parameter, as many tasks performed by coordinators require the execution of various algorithms.

Available CPU resources may be limited even on powerful systems, as MMVEs place high load on the system through calculation of physics, artificial intelligence calculation or real-time 3D graphics. This makes both the CPU hardware (e.g., clock rate and number of cores) and the current CPU load relevant parameters.

The cost of determining both CPU hardware and current CPU load is low, as these values can usually be determined by a simple request to the operating system, which maintains these values. Consequently, reliability is high. In theory CPU hardware has both high variance and a high magnitude of change. The value only changes when the CPU of a system is replaced, which commonly happens in order to install a more powerful unit, thus resulting in high magnitude and variance. In practice however, the CPU is rarely changed. This means the frequency of changes is so low, that fluctuation can be ignored in practice. The current CPU load on the other hand has high variance, high magnitude of changes and changes frequently. While the MMVE itself can be assumed to generate consistent CPU load, the value can fluctuate due to other software running on the system.

2) Memory: The additional functions of a coordinator also require a certain amount memory, for example to save routing information. The total and available memory can easily be determined through the operating system. This results in high reliability and low cost. As with the CPU, the total amount of memory in a system will rarely change, but the available memory can fluctuate due to additional software running on the system, resulting in high variance and frequent changes with high magnitude.

3) Storage Space: Another possible requirement for a suitable coordinator is storage space, e.g. on a hard disk. This storage space could for example be used to store game content (3D models, textures, etc.) which is distributed to the peers by the coordinator. The available storage space can be requested via the operating system, which places limited load on CPU and the disk. Cost is therefore low, with high reliability. Variance is high, as a previously empty disk can completely fill up over time. Regular changes to the available disk space are also made, as data is written to or removed from the disk. These changes are usually small however, particularly in relation to the total disk space, resulting in a low magnitude of changes.

B. User Behaviour

Parameters in this category are based on the behaviour of the peer's user. The parameters are stability and reputation.

1) Stability: Replacing coordinators is associated with overhead. As such it is desirable for a peer to remain coordinator for long periods of time. Using information on the user's past and current behaviour, it is possible to make predictions about his session time [15]. For example, some users have longer average session times than others. In addition, users which have just logged into the system are more likely to leave the system soon, than users which have already spent a certain time in the system. Peers which tend to leave the system gracefully, i.e. which are shut down by the user as intended by the developers, are also preferable over peers which often leave the system non-gracefully, e.g., due to system crashes. This is due to the fact that a non-graceful exit from the system can increase the overhead of replacing the coordinator, or lead to degraded performance of the system.

This type of behaviour is referred to as stability. The more stable a peer is, the more likely it is to remain in the system for a long period of time and leave it gracefully afterwards. Stability is a parameter with low reliability. Even when a long session time is predicted for a particular peer, it may still leave the system at any time.

As the value of this parameter is based mostly on a history of previous behaviour, this value adapts slowly to the changing behaviour of the user. As such the magnitude of changes is low. As the value changes only when new information is added to the history, the frequency of changes of also low. Variance can be large however, as a user's behaviour changes over time. Cost is low, because session time can be measured with minimal overhead.

2) Reputation: Reputation mechanisms are used discourage or encourage certain user behaviour, such as cheating. Peers give each other positive or negative feedback after certain types of interaction, which is combined to create a reputation value for each peer. This can be done algorithmically or through user feedback. Some reputation values can be useful for determining a peer's suitability as coordinator. For example, a user which has accrued a reputation for cheating may be more likely to present a threat to the integrity of the MMVE. Handing such a peer additional responsibilities can therefore expose the system to further cheating attempts from the user.

Depending on the number of interactions rated by the reputation system and the number of peers involved in these interactions, the creation of the reputation feedback can result in high cost. The reputation system creates many small changes to a peer's reputation, resulting in high frequency but low magnitude of changes. Variance can be high however, as a user's behaviour changes over time.

C. Network Access

Parameters in this category are based on the type and state of the user's Internet access. The parameters are bandwidth, Network Delay and Connectivity Restrictions.

1) Bandwidth: Coordinators incur communication overhead. This makes bandwidth one of the key parameters for peer utility. There are three aspects that need to considered, when looking at bandwidth: a) maximum bandwidth b) overall available bandwidth c) available bandwidth to a specific peer. For each of these aspects upstream and downstream bandwidth need to be considered separately, as bandwidth is often asymmetrical, with upstream bandwidth being the more limited resource.

The maximum bandwidth is the theoretical maximum bandwidth available to the peer. It is determined by the type of Internet connection of the peer's system. Due to several factors, such as sharing the connection with other systems, bandwidth usage by other applications on the peer's system and under-provisioning by the Internet service provider during peak hours, the maximum bandwidth cannot always be achieved.

Available bandwidth refers to the actual bandwidth that is currently available to the peer for coordinator functions. The available bandwidth is the maximum bandwidth minus all currently active bandwidth reducing factors. Note that this includes the bandwidth used by the peer for non-coordinator related functions in the MMVE.

It should be noted, that while a peer may have high available bandwidth, it does not follow that this bandwidth is necessarily available to all other peers in the system. The actual data transfer rate from one peer to another depends on the available bandwidth of both peers as well as any possible bottlenecks on the network route between the to peers. As such, even if a peer has high available bandwidth it may still have a low utility with regard to bandwidth, if its position in the network topology means that data sent to or received by it passes through a bandwidth bottleneck. It can therefore be necessary to consider the actual available bandwidth to a specific peer.

When considering bandwidth as a parameter, it is important to consider upstream and downstream bandwidth separately. Depending on the coordinator's functions, incoming and outgoing traffic requirements may be different and many consumer Internet connections have asymmetric bandwidth, with significantly lower maximum upstream bandwidth than maximum downstream bandwidth. High bandwidth usage in one direction can also negatively affect the available bandwidth in the other direction, which must be taken into account when determining available bandwidth.

The cost of determining maximum or available bandwidth can be high, as some active bandwidth measuring algorithms send out the maximum possible bandwidth for a period of time. Despite this, based on various factors such as network bottlenecks or a shared Internet access, the reliability of these algorithms is generally low.

While the maximum bandwidth only changes, when the user switches his type of Internet access, the available bandwidth can change frequently with large magnitude of changes, due to other applications or systems using the same Internet access. If these factors are present, variance is also high.

2) Delay: As MMVEs are highly interactive systems, delay is an important parameter, particularly if the coordinator is used for the propagation of event updates in the virtual world. High network delay results in a reduction of the level of interactivity of the MMVE, which in the worst case can make participation in the virtual environment impossible.

Delay between two individual peers is easy to measure. Cost for a single delay measurement is low. However, due to the nature of an MMVE, a single delay measurement is of limited use to determine peer utility. Just because a peer has low delay to one other peer, does not mean it will have low delay to all peers which it serves as coordinator. Due to the nature of MMVEs, the coordinator often cannot choose which peers it serves, as this is determined by neighbourhood in the virtual world. Consequently a single coordinator may serve peers which have relatively high delay to it, as they are large distance from the peer in the network topology.

Individual delay measurements are reliable, however packet delay variation (PDV) can cause problems. Sometimes called jitter, PDV is the difference in end-to-end delay in a sequence of packets between two hosts. When jitter is high, delay exhibits high variance, with frequent changes of large magnitude. Due to this, as well as the fact that individual delay measurements are not meaningful and massive numbers of delay measurements are cost-prohibitive, it is difficult to determine delay reliably and with low cost.

3) Connectivity Restrictions: Many peers in a P2P-MMVE are behind a firewall or network address translation (NAT), often in the form of a residential gateway (e.g., a wireless router/access point). Any P2P-MMVE must be able to establish connections among such peers, using techniques like hole punching and connection reversal. This generates overhead however, and delays the establishment of the connections.

In addition many residential gateways are not designed with heavy network activity in mind, and commonly limit the number of connections that can be established at any given time. Too many connections can crash residential gateways. As such it is preferable to select coordinators which are not behind a residential gateway or NAT and firewalls. However, an existing firewall may also be considered to be a positive factor, as it makes the coordinator safer against security attacks. While it is difficult to determine the specific type of residential gateway, it is possible to detect NAT and firewalls. This detection can be done reliably with limited overhead.

Once these values have been established, they tend to remain the same, as users rarely change their residential gateway hardware or configuration. In practice we can therefore consider them to have low variance, low frequency of changes and low magnitude of change.

D. Parameter Summary

As we have seen, parameters based on the hardware of the peer's system, can often be determined reliably and with low cost. High fluctuation of the values is a problem, but this is offset by the low cost of the measurements, which means values can be re-measured at a high frequency. The high fluctuation must be considered however, so that outliers do not have have excessive influence on the peer's utility.

Determining parameter values based on user behaviour is more difficult. As the values are generally predictions based on past behaviour, they suffer from a lack of reliability. If peers share information on each other's behaviour, such as in a reputation system, the cost can also become a major factor.

Network access is the probably the most relevant of the three categories As highly interactive distributed systems, P2P-MMVEs key functionalities are based on the Internet. The large scale of the system makes usable delay measurements difficult and determining bandwidth values is inefficient and unreliable in client/server systems and even more so in P2P systems. Measuring these parameters reliably and with low cost remains a significant research challenge.

In the following section we will look at our own P2P-MMVE system and show that our specific coordinator selection is not solved by existing approaches.

V. Coordinator Selection in peers@play

In the peers@play project, we have built a coordinator-based P2P-MMVE. In our system coordinators are primarily used for update propagation [1]. Peers suitable as coordinators should have available CPU resources, high stability, high available bandwidth and low network delay.

Additional CPU resources are used for running our interest management. High stability is advantageous due to the overhead associated with coordinator selection and handover. High available bandwidth is required, as the coordinator must propagate additional updates. Finally, to ensure high interactivity, coordinators should have low delay to their peers.

We have found that out of these four parameters, we can only determine the CPU power reliably and at low cost. While we have a simple solution for stability, we currently do not have satisfactory solutions for bandwidth and delay. In the following we will very briefly discuss our solutions for CPU and stability as well as the problems we encountered trying to determine bandwidth and delay.

To determine a value for CPU, we get information on the current use of CPU resources from the operating system and use a history function to deal with the high fluctuation of the result. In addition, we prefer peers with more CPU cores.

To determine stability, we use a history of a peer's session length. In addition, we avoid peers whose current session length has not yet exceeded a certain threshold.While this simple approach has relatively low cost, it takes several sessions for the reliability of the value to become acceptable. Given the usage pattern of MMVEs it can therefore take days before we can get a reliable value for any given peer.

The available bandwidth is the most important parameter for our approach. A solution which suits our needs would have to be low-cost, provide results quickly and deal well with high fluctuation. We are currently aware of only two approaches to determine available bandwidth in a P2P system ([16] [17]), neither of which fulfils these requirements. We have therefore started research into our own solution.

Finally, there are a number of approaches to determine the expected network delay in P2P systems (e.g. [18]). Unfortunately, all of these approaches assume that a peer can select the coordinator it wants to connect to. However, in zone-based MMVEs such as peers@play, peers must connect to a pre-determined coordinator. This coordinator depends on their avatar's current location in the virtual world. Selecting a suitable coordinator which can provide low delay to as many peers as possible, is a problem we currently consider unsolved. As a consequence, we are ignoring this parameter during our coordinator selection.

In conclusion, in order to deploy our system in a real-world scenario, additional research is required. We believe the same is true for most other coordinator-based approaches as well.

VI. Related Work

A number of MMVE systems use the topology of the underlying P2P overlay network to select coordinators. In [2] each region is assigned a key from the Pastry key space. The peer whose ID is the closest to the region ID becomes the coordinator. In [7], each peer manages those regions that are numerically nearest its ID in the DHT. [9] uses JXTA to organize peers in groups: if a region needs a new coordinator, a peer from the group is used.

Some approaches use the avatar position in the virtual environment to select coordinators. A peer is selected if there is no coordinator in the user's zone [3] [5], or if the user position is the closest of the users to the geometric center of the region [14]. Even though all of these approaches put additional load on the coordinators, they are not selected based on specific suitability.

Some approaches note that their coordinators have higher requirements than usual, e.g. regarding bandwidth [6] [4] [10], CPU power [6], delay [4], trustworthiness [10], stability [10] and storage capacity [8]. However none of these approaches provide specific techniques for determining the values for these parameters. While the ability to determine suitable peers is crucial to them, they explicitly or implicitly assume their ability to do so.

VII. Conclusion and Future Work

In this paper we have argued that coordinator selection in P2P-MMVE is not solved. We believe that it is one of the few remaining challenges for the real-world feasibility of P2P-MMVEs. We have provided an analysis of relevant parameters and the challenges associated with determining them. We hope that this analysis will inspire a robust discussion about the necessary research into this area.

References

[1] R. Sueselbeck, G. Schiele, S. Seitz, and C. Becker, "Adaptive update propagation for low-latency massively multi-user virtual environments," in *Proceedings of 18th International Conference on Computer Communications and Networks, 2009. ICCCN 2009.*

[2] B. Knutsson, H. Lu, W. Xu, and B. Hopkins, "Peer-to-peer support for massively multiplayer games," in *Proceedings of the 23rd Conference of the IEEE Communications Society (INFOCOM)*, 2004.

[3] T. Iimura, H. Hazeyama, and Y. Kadobayashi, "Zoned federation of game servers: a peer-to-peer approach to scalable multi-player online games," in *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, 2004.

[4] C. GauthierDickey, V. Lo, and D. Zappala, "Using n-trees for scalable event ordering in peer-to-peer games," in *Proceedings of the 15th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, 2005.

[5] A. Yu and S. T. Vuong, "Mopar: a mobile peer-to-peer overlay architecture for interest management of massively multiplayer online games," in *Proceedings of the international workshop on Network and operating systems support for digital audio and video*, 2005.

[6] S. Yamamoto, Y. Murata, K. Yasumoto, and M. Ito, "A distributed event delivery method with load balancing for mmorpg," in *Proc. of 4th ACM SIGCOMM workshop on Network and system support for games*, 2005.

[7] A. Chen and R. R. Muntz, "Peer clustering: A hybrid approach to distributed virtual environments," in *Proc. Netgames*, 2006.

[8] D. T. Ahmed, S. Shirmohammadi, and J. C. de Oliveira, "A novel method for supporting massively multi-user virtual environments," in *Proceedings of the IEEE Workshop on Haptic Audio Visual Environments and their Applications*, 2006.

[9] H.-H. Lee and C.-H. Sun, "Load-balancing for peer-to-peer networked virtual environment," in *Proc. Netgames*, 2006.

[10] S.-Y. Hu, C. Wu, E. Buyukkaya, C.-H. Chien, T.-H. Lin, M. Abdallah, J.-R. Jiang, and K.-T. Chen, "A spatial publish subscribe overlay for massively multiuser virtual environments," in *Proc. 2010 International Conference on Electronics and Information Engineering*, 2010.

[11] T. Hampel, T. Bopp, and R. Hinn, "A peer-to-peer architecture for massive multiplayer online games," in *Proceedings of 5th ACM SIGCOMM workshop on Network and system support for games*, 2006.

[12] L. Itzel, V. Tuttlies, G. Schiele, and C. Becker, "Consistency management for interactive peer-to-peer-based systems," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools '10, 2010.

[13] S.-Y. Hu, S.-C. Chang, and J.-R. Jiang, "Voronoi state management for peer-to-peer massively multiplayer online games," in *Proc. 4th IEEE Intl. Workshop on Networking Issues in Multimedia Entertainment*, 2008.

[14] P. Morillo, W. Moncho, J. Orduna, and J. Duato, "Providing full awareness to distributed virtual environments based on peer-to-peer architectures," in *Lecture Notes on Computer Science*, 2006.

[15] D. Pittman and C. GauthierDickey, "A measurement study of virtual populations in massively multiplayer online games," in *Proceedings of the 6th ACM SIGCOMM workshop on Network and System Support for Games, NetGames '07*, 2007.

[16] R. Snader and N. Borisov, "Eigenspeed: secure peer-to-peer bandwidth evaluation," in *Proceedings of the 8th international Workshop on Peer-to-Peer systems*, ser. IPTPS'09, 2009.

[17] J. Douceur, J. Mickens, T. Moscibroda, and D. Panigrahi, "Collaborative measurements of upload speeds in p2p systems," in *INFOCOM, 2010 Proceedings IEEE*, 2010, pp. 1 –9.

[18] S. Agarwal and J. R. Lorch, "Matchmaking for online games and other latency-sensitive p2p systems," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, ser. SIGCOMM '09. New York, NY, USA: ACM, 2009, pp. 315–326.