The Near-Term Feasibility of P2P MMOG's

John L. Miller[1,2]
[1]Microsoft Research, Cambridge
Cambridge, United Kingdom
+44 1223 479813
johnmil@microsoft.com

Jon Crowcroft[2]
[2]University of Cambridge Computer Laboratory
Cambridge, United Kingdom
+44 1223 763633
jac22@cl.cam.ac.uk

Abstract-- **Broadly deployed distributed virtual environments (DVE's) are based upon client-server architectures. Significant research over the last decade proposes a variety of distributed topologies for message propagation to enhance scalability and performance. We ran simulations using traces from World of Warcraft (WoW) and typical broadband speeds, and found that DVE's such as WoW could not employ P2P message propagation schemes. We found pure P2P pub-sub solutions would occasionally saturate residential broadband connections, and would result in average latency more than 10x greater than client-server solutions. To our surprise, we discovered message aggregation before transmission can not only reduce bandwidth requirements, but also reduce latency in both client-server and P2P message propagation schemes.**

Keywords - World of Warcraft, P2P, Distributed Virtual Environments, Massively Multiplayer Online Games, Latency

I. INTRODUCTION

Ten years of significant research on peer-to-peer distributed virtual environments (P2P DVE's) has yielded many new techniques, yet few have been adopted by deployed systems. One possibility – which we believe this paper confirms - is that these technologies may not be deployable given typical residential broadband infrastructure.

Nearly every aspect of running a reliable DVE becomes more challenging in a P2P architecture. While there are any number of practical challenges to adopting and deploying a peer-to-peer protocol, we ask a more fundamental question. Ignoring all other issues, can existing residential network infrastructure support the bandwidth requirements of a P2P DVE?

We gathered and analyzed network traces from the world's most popular DVE, a massively multiplayer online game (MMOG) called World of Warcraft (WoW). We used these traces as the basis of a workload for simulating client-server and an idealized peer-to-peer publisher / subscriber solution. We were able to characterize both absolute and relative performance of both types of messaging solutions, and found even an idealized P2P solution to perform too poorly to deploy.

The remainder of this paper describes our investigation. First, we provide background relevant to WoW and the broadband bandwidth model we used. Next we discuss our methodology for creating workload traces, and our simulator's behavior. We analyze the gross attributes of our parsed WoW workloads, and apply them to both client-server and P2P pub-sub message schemes. Finally, we present conclusions and directions for further research.

II. BACKGROUND

This section provides information on World of Warcraft and its main characteristics. It also gives a brief summary of the OfCom-provided residential broadband statistics used in our bandwidth model.

A. World of Warcraft

World of Warcraft (WoW) is the most popular Massively Multiplayer Online Game in history. With more than 11 million subscribers worldwide, WoW has the majority of market share for all MMOG's: 62% as of 2008 [1].

Player avatar activities in WoW can be divided into five categories, as proposed by [2]:

- **Questing**. The avatar interacts with the game environment and AI-controlled avatars called NPC's.
- **Capitals**. This category is a subset of the *trading* category proposed in [2], and consists of time spent in the game's heavily populated capital cities.
- **PVP**. Player versus player combat. We use this category for inter-player battles in zones called *Arenas* or *Battlegrounds*. There are six unique battlegrounds.
- **Dungeons**. Small, scenario-driven experiences where up to five avatars work together to kill a series of powerful NPC's known as 'bosses'. There are 56 unique dungeons.
- **Raids**. A large dungeon. Instead of 2 – 5 bosses, raids have between 1 and 12. They are designed for 10-40 players and may include hundreds of NPC's. There are 24 unique raids.

[3] examines WoW server populations using WoW extensibility. The authors found DVE simulation workloads were unrealistic. They found peak server populations are typically five times their minimum population, underscoring the need to model peak behaviors.

[4] analyzes avatar movement in the Arathi Basin battleground. It proposes a mechanism for obtaining game positions from avatar traces. We extend that work, parsing several other types of messages, and obtaining both player and NPC messages and positions.

Evaluation of proposed DVE systems often use synthetic workloads based on previous research, or on a model generated by the evaluators. For example, [5] compares three different categories of DVE infrastructure using a synthetic workload based upon an average session time of 100 minutes. Avatars in that evaluation are simulated using a combination group and waypoint model, where groups of simulated avatars agree on a next point to visit, and move there together. Several other frameworks [6] [7] [8] [9] assume movement and arrival / departure properties of participants without any obvious experimental basis. The assumptions used in all these cases are logical, but without firm experimental grounding, and often contradict our measurements from WoW.

B. Broadband speeds

Our analysis examines behavior of different topologies involving residential nodes. It requires a realistic consumer broadband model.

Broadband adoption has been growing throughout much of the world. We based our model upon UK broadband speeds.

Actual residential broadband speeds are on average significantly slower than their advertised speeds. Ofcom - the communications regulator for the UK - recently published a residential broadband study [10]. The study characterizes ISP's covering 90% of the broadband-using UK population. Salient statistics are summarized below:

- Ofcom divided subscriptions into three categories: 2 Mbps or less, > 2 Mbps and <= 8 Mbps, and > 8 Mbps advertised download speed. Relevant statistics are summarized in Table 1.
- Round-trip latency as measured to UK servers was on average below 55 ms, with peak average latencies (at high-traffic times of day) of up to approximately 110 ms.
- Connections encountered low jitter, typically less than 6 ms (9% of the 55ms RTT latency).
- Packet loss averaged less than half a percent.

Our evaluation of P2P feasibility does not model jitter or packet loss. However, it makes use of the population percentages, and upload and download speeds given above.

The next section describes our overall approach for analyzing World of Warcraft captures and making simulation traces from them.

III. METHODOLOGY

Our goal was to determine whether a P2P message propagation scheme could propagate the messages required for an MMOG to operate. We had to answer two questions. First, which avatars receive which messages? Second, which network attributes are most relevant to our simulation?

MMOG's typically strive to propagate messages only to clients affected by the message. For example, if player X's avatar is moving and within visual range of player Y's avatar, then X's DVE instance needs a copy of Y's movement messages.

Most DVE messages reflect a state change initiated by and centered on a single avatar. Some messages have a broader effect – such as a command to render rain in a large zone – but these commands are relatively rare. Messages are received by a superset of the avatars affected by that message. For example, any avatars within visual range of a spell being cast need to know about the spell, even if they are not facing the caster. WoW Avatar visible range is usually between 250 and 500 yards. Interaction range is shorter, about 5 yards for melee and trading, or 30 yards for most ranged interactions.

For an ideal P2P messaging scheme, we make the following simplifying assumptions.

1. All clients can communicate with each other
2. Clients have perfect knowledge of other clients entering and leaving their interaction and visual range
3. There is no overhead associated with setting up client network links or with tracking remote nodes beyond what is normally propagated by the DVE.
4. There is no additional security overhead, nor other costs usually associated with P2P message propagation.
5. There is no packet loss or jitter.
6. The underlying transport is TCP. The only simulation impact of this choice is message framing overhead, and the TCP window enforcing a limit on outstanding bytes for a given connection.

These assumptions reduce traffic requirements. If the network can't support this simplified system, then it certainly can't support a more realistic (and complex) solution.

Table 1 - Client Node Bandwidth Model

Advertised Speed	Subscriber Percent	Down (kbps)	Up (kbps)
<= 2 Mbps	29%	1700	280
2 to 8 Mbps	57%	3900	420
> 8 Mbps	14%	9300	580

Our P2P pub-sub message propagation scheme implements a pure P2P model. Each message is transmitted by the originating client directly to each subscriber requiring a message copy.

A. World of Warcraft Network Attributes

WoW uses a proprietary client-server protocol to communicate between DVE nodes. Most data between a WoW client and its server is exchanged over a single TCP connection. Most traffic results from client node actions. The acting client node sends a message to the server, which makes appropriate DVE state changes, then forwards the message or related updates to any other affected nodes.

Each client DVE instance does an advisory check of local user inputs and state change requests before submitting them to the server. If the server accepts a request as legal based on its world state, the message (or resulting update) is propagated to clients to whom it may be relevant, sometimes including the client who submitted the message. If the action was rejected, the server replies to the requester, and that requesting client reverts any relevant state.

WoW network messages are small, 36 bytes on average [2], with some as small as 4 bytes. During active periods a client can send more than 10 messages per second. Conversely, in idle periods the client may go minutes between sending messages. Messages consist of an encrypted header and an optional unencrypted body. Messages can be combined within a single TCP packet, and if needed can span TCP packet boundaries. During periods of intense local activity clients may receive bundles of many messages several MTU's in size.

Most messages include a globally unique avatar identifier (guid) identifying the party initiating and / or affected by the message. Other common fields include X, Y, and Z position in yards, facing information, and game time.

Since message headers are encrypted, we devised heuristics to identify a few of the more complex (easier to identify) and useful message types. Like WoW, we propagate messages based on virtual world proximity, so position messages are our highest priority. After parsing a candidate list of move messages, we revert incorrectly parsed messages by comparing their contents against median values. Correctly parsed sequences of messages have similar values in many fields (such as position and game time), whereas incorrectly parsed messages are more likely to have divergent values.

B. Simulation Trace Generation

Our simulation traces are based upon processed WoW network captures. Each capture is between 5 and 120 minutes long. In some cases the captures were from a single client instance, while in others a pair was used to extend the region of visual information.

Before parsing messages, we divided the TCP payloads of the trace into *blobs*, aggregating adjacent packet payloads which were clearly part of the same blob. For example, two MTU-sized packets followed by a third smaller packet would be combined into a single blob.

We iterated over the blobs, attempting to parse message types from most to least restrictive. After parsing each message type,

heuristics were used to revert clearly erroneous parsing. For example, if the guid didn't meet structural expectations, or didn't appear frequently enough in the parsed trace, the message was considered invalid and reverted.

Once all messages were parsed and filtered, we searched through any remaining blobs for known guids. Whenever we found a guid, we assumed it was at a common offset for guids within messages, and considered the message as starting an appropriate number of bytes before the guid. We called this process *attribution*. The message was assumed to continue to the end of the blob, or to the start of the next attribution, whichever was shorter. The logic behind this strategy is as follows: identifying a party involved in a propagated transaction almost always means the message originated within interaction distance (30 yards) of that party. Since interaction between avatars is typically limited to an area much smaller than – and fully contained within – the maximum visual AoI, a slight variance in the propagation center won't significantly affect the receiving audience. This assumption allows us to assign the majority of traffic we don't successfully parse in a trace to specific avatars for origination. While not strictly correct, we believe it provides a good approximation. As data later shows, using this strategy we're able to successfully parse or attribute more than 90% of message bytes.

Earlier we described the five categories of play within WoW. We captured data in all five categories to gain a better understanding of difference between them, and to ensure we evaluated the network topologies in all styles of play.

Fidelity of the captures and their overall utility varied by category:

1. **Questing**. We captured and parsed solo play with both highest-level characters and characters advancing in the game. Solo PVE usually involves significant travel in sparse areas. Our captures are limited to the solo player's AoI, and are low fidelity from a simulation perspective because of a lack of context.
2. **Capitals**. Capitals are heavily populated, with a high degree of transience. Most avatars are quiescent, performing simple maintenance activities or waiting for game events. We captured traces in Dalaran, the most popular capital city, with two avatars strategically placed to capture most of the city's messages. Dalaran's population ranges from dozens to hundreds of players, with turnover as high as thousands of unique avatars per hour. We have good fidelity captures for this category.
3. **Dungeons**. In dungeons a player avatar explicitly works with up to four other avatars to complete a bounded objective such as killing a series of NPC's. Players stay within interaction distance of each other, so the capture contains all relevant information.
4. **Raids**. A raid is a special type of dungeon for 10 to 40 avatars rather than 5. Other than scale, they're similar to dungeons, and have good capture fidelity.
5. **PVP Battlegrounds**. We captured data in two of the games' six battlegrounds, Arathi Basin and Wintergrasp. We have high fidelity captures for Arathi Basin, as our two instrumented nodes were sufficient to cover most of the battleground. Wintergrasp is larger, and we were only able to capture information at the main hotspots. This is sufficient to characterize the highest activity areas centers, but misses smaller battles located away from the hotspots. Despite limitations of the Wintergrasp captures, they are interesting because of their scale, often more than a hundred participants, with the majority mutually interacting.

After parsing and processing this data as described above, we had a series of traces appropriate for simulation. The next section describes the simulator's basis. Summary statistics for data capture and parsing can be found in section IV.

C. Simulator

We developed a medium-fidelity packet-level network simulation, which provides a reasonable approximation of internet behavior for our scenarios.

- Our simulation assumes TCP / IP as the message transport, and uses a packet-based model for propagating data.
- Nodes are assigned 'network positions' on a 2D plane. Latency is the Cartesian distance between two nodes' network positions.
- Messages have framing overhead (TCP and IP). Large messages are divided into packets based upon MTU.
- We account for first- and last- hop bandwidth limitations, but as a simplifying assumption treat capacity between these two points as infinite.

The simulator propagates messages at three levels: The TCP/IP stack, the network adapter (NIC), and the last-hop router. Each is outlined in more detail below.

1) TCP / IP level. The TCP/IP level mirrors behavior of relevant portions of a typical Windows TCP/IP stack. It accepts messages for transmission, applies TCP windowing, aggregates messages, and forwards them to the NIC queue to simulate outbound transit behavior. We use a TCP Window size of 17 kB, the default TCP Window size for Windows XP.

Message transmission through the TCP/IP stack is simulated in the following sequence:

1. Locally originated messages are entered into a src→dest keyed outbound queue.
2. Every millisecond the state of the src→dest connection is checked to see how many packets have left the IP stack at src, but have not been fully received at dest. If the payload bytes exceed the TCP window, the messages remain buffered.
3. Otherwise, the TCP/IP stack transfers enough messages to the outbound NIC level to fill the TCP window.

2) NIC level. The NIC level of the simulation handles inbound and outbound packets on a first-come, first-serve basis. Packet transmission is simulated in millisecond time slices, subject to node bandwidth limits. When packets finish transit, they are removed from the NIC-level queue and passed to the next layer, either the TCP/IP queue for inbound packets, or the remote 'last-hop router' queue for outbound messages.

When a packet is transferred to a 'last-hop router' queue, it is time-stamped with the current simulation time plus any inter-node latency. In other words, the packet is marked for future delivery.

3) Last-hop router. Internet latency and inbound NIC bandwidth are both modeled using the 'last-hop router' before the destination node.

The last-hop router has a priority queue with zero or more packets, ordered by their projected arrival time. Once a packet's arrival time passes, the last-hop router moves it from the priority queue to a host transmission queue. The packets are moved into the inbound NIC at the destination host on a FIFO basis, using the time slice bandwidth budget. Whenever a packet finishes

transmission, it is passed into the receiving node's NIC queue, where it then bubbles up the stack as a received WoW message, and is processed.

D. Topology Choices and Metrics

DVE topologies can be divided into three broad categories:

1. Pure client-server. The server has 'perfect' information about the location of every avatar, and uses this as the basis for propagating messages. This model is used by WoW.
2. Region-based pub-sub. The game space is divided into regions. A subset of nodes propagate messages for each region.
3. Avatar-based "P2P" pub-sub. Avatars subscribe to event streams for other avatars within a certain distance of themselves, i.e. within their Area of Interest (AoI).

The main goal of these schemes is to ensure each node gets its messages in a timely fashion. 'Timely' depends upon the DVE. WoW defines three latency thresholds: <= 300 ms latency is *good*, <= 600 ms is *adequate*, >600 ms is *poor*. Other DVE's have documented acceptable latency tolerance as low as 200ms and as high as 1250 ms. Lower latency is almost always better.

E. Simulation Characteristics

We performed three sets of simulations. For each set, we ran all our simulation traces on both a pure client-server topology, and a P2P pub-sub topology.

For each simulation, we set the following attributes:

1. Weighted random assignment of client node bandwidth, based upon OfCom statistics, as shown in Table 1.
2. Random assignment of 2D Cartesian network coordinates to model inter-node latency, with a maximum inter-node latency of 90 ms and an average inter-node latency of 45 ms.
3. For client-server simulations,
   a. Symmetric server bandwidth (if a server is present) of 1 Gbps
   b. Server is assigned the center of the latency modeling space to mimic "optimal network placement" of the server.
4. AoI of 250 yards. All nodes subscribed to events from any other node within their AoI. If events originate at a non-node (e.g. an NPC), they are propagated by the closest node.
5. TCP window size of 17 kB.

The three simulation sets had a single parameter varied between them, the message aggregation interval. For each interval, all messages to be transmitted over a given TCP connection were bundled up and an aggregated message – divided into multiple IP packets if required - transmitted. The first run had an aggregation interval of 0, meaning messages were immediately processed without aggregation. We subsequently used aggregation intervals of 1ms and of 50ms. Surprisingly the results between 1ms and 50ms were not substantially different from each other, so we will only discuss the difference between the 0 ms and 1 ms aggregation intervals.

For both client-server models and P2P models, we assume perfect knowledge of subscribers, based upon the actual position of avatars at the message origination time.

The next section describes our results. These include our success in understanding and parsing messages for each of the WoW activity categories, and results of our simulation runs.

IV. RESULTS

Table 2 shows the percentage of payload data we successfully parsed and attributed by play category. 'Parsed' is the data fraction parsed into messages we understood. 'Attributed' is the data fraction we were able to attribute, but not parse. 'Discarded' is any remaining bytes, since we could not determine the propagation AoI.

The largest proportion of unrecognized, unattributed data in traces came from questing, nearly half of all questing data captured. The most action-intensive categories, Raiding and PVP, had less than 10% of data discarded, and so provide realistic simulation traces. Table 3 shows the aggregate duration of captured data, in minutes.

While trading time appears to be insignificant (just over 3 hours of data captured), it is high fidelity data from a relatively uniformly behaved area (per time of day).

We consider Capitals, Raiding, and Dungeons key scenarios. Our workload trace attributes more than 90% of traffic bytes in two of these cases, and so we believe it is a good basis for simulation.

A. Simulation Results

We had two significant results in our simulation.

1. **P2P pub-sub is not feasible today**. For most scenarios we found nodes occasionally fell behind, in some cases receiving messages minutes after their origination. This is unacceptable for DVE's such as World of Warcraft.
2. **Message Aggregation reduces P2P pub-sub latency**. Common wisdom and published speculation says delaying message transmission is purely harmful in latency-sensitive applications. We found otherwise, especially in bandwidth-constrained scenarios.

1) Feasibility of P2P pub-sub. Our goal was to determine whether peer-to-peer MMOG's are possible with today's infrastructure. Our conclusion is that they are not. This section provides supporting evidence for that conclusion.

To evaluate our hypothesis, we used World of Warcraft data and our own simulator. Our analysis indicates the WoW protocol is quite efficient. Upload and download bandwidth consumption in an older version was found to be on average 2.1 kbps upload, and 6.9 kbps download [11]. [2] analyzed a later version of World of Warcraft and found that upload bandwidth was usually 10 kbps or less, and average download bandwidth varied from less than 10 kbps for questing, to approximately 50 kbps for raiding. Our observations on WoW version 3 are consistent with these latter numbers.

Table 2 - Parse Percentages

	Parsed	Attributed	Discarded
Questing	25%	30%	45%
Capital	37%	39%	24%
Dungeon	31%	58%	11%
Raid	26%	66%	8%
PVP	42%	51%	7%

Table 3 - Minutes of data Captured

Category	Total Minutes	Average Per Capture
Questing	1339.07	44.64
Capital	204.54	29.22
Dungeon	1136.80	27.73
Raid	703.14	31.96
PVP	310.55	16.34

As mentioned earlier, we ran simulations across traces captured from each of the five categories of play, with a message propagation radius of 250 yards, at the lower end of our AoI estimate for WoW.

Running our simulation against a client-server architecture, we found results consistent with our first-hand experience in terms of message latency. Results are summarized in Table 4. The first two columns present the average upload and download bandwidth consumed by each client by category. The third column shows overall latency in seconds, e.g. 0.046 seconds which is 46 milliseconds average. The final column shows the average peak latency. In other words, if you calculate the greatest latency during each player session in a given category, then average those values together, that is the 'peak' value shown in this column.

We configured an average inter-node latency of 45 ms. The server is placed in the center of the 'network latency' map, making node-server latency on average 22.5 ms.

The same simulation run with a P2P pub-sub scheme did markedly worse, as shown in Table 5. In the pub-sub model, we assume peers are responsible for uploading not only their own messages, but for modeling and transmitting any messages on behalf of NPC's closest to them. This increases peer upload bandwidth requirements, but reduces download bandwidth requirements as they are synthesizing some NPC messages rather than receiving them from a server.

Comparing the two tables, we find that the average upload bandwidth required per publisher is a factor of 20 greater than client-server overall, and in some cases, a factor of nearly 100. Download dropped slightly, while latencies rose to unacceptable levels for all scenarios except *Questing*, which requires little message propagation. In addition to significant latency, we found peak upload bandwidth requirements exceeded capacity in most cases, for at least part of each trace. This was caused by a large number of avatars interacting near each other. In raids this could consist of 25 player avatars and 100 NPC's simultaneously moving and using special abilities. In PVP battlegrounds the peaks were especially obvious in Wintergrasp during battles at one of the objective fortresses' walls. In the capital, it was a function of how many users were online at a given time. The more avatars online, the greater the number of event subscribers.

WoW is – in the authors opinions – a well-designed game in terms of network efficiency. The experience has been carefully crafted to require minimal per-node resources. Even with low per-node network requirements, the simple need to exchange information frequently between adjacent nodes makes it infeasible to make a responsive pure P2P MMOG, at least with today's bandwidth limitations. There is hope for the future. First, broadband speeds are increasing as new technologies are rolled out with ever-increasing speeds. Second, there are mitigation strategies such as avatar behavior modeling [12] which can be explored to reduce communication frequency and size requirements.

Table 4 - Average Client-Server Simulation Results

	Up (kbps)	Down (kbps)	Average Lat. (S)	Peak Lat. (S)
Capital	1.8	84.3	0.047	0.269
Instance	4.1	35.7	0.046	0.342
PVP	3.2	48.8	0.049	0.150
Quest	2.7	5.5	0.036	0.079
Raid	5.2	80.4	0.051	0.658
Average	3.4	51.0	0.046	0.299

Table 5 - Average P2P Pub-Sub Simulation Results

	Up (kbps)	Down (kbps)	Average Lat. (S)	Peak Lat. (S)
Capital	167.2	79.2	4.467	119.697
Instance	38.6	32.7	1.163	12.763
PVP	83.2	48.3	1.420	24.711
Quest	5.8	4.4	0.033	0.121
Raid	102.2	77.6	9.070	92.708
Average	79.4	48.4	3.231	50.000

Table 6 - Client-Server, 1 ms Aggregation Changes

	Up (kbps)	Up Peak (kbps)	Down (kbps)	Down Peak (kbps)	Avg Lat. (S)	Peak Lat. (S)
Questing	-9.0%	-20.3%	-14.3%	-26.0%	0.0%	-2.2%
Capital	-25.0%	-12.1%	-31.8%	-26.6%	0.8%	-4.3%
Dungeon	-17.6%	-22.8%	-20.4%	-28.4%	0.6%	-3.7%
Raid	-6.1%	-10.6%	-17.7%	-12.6%	0.2%	-3.5%
PVP	-31.7%	-16.6%	-38.6%	-34.1%	-0.3%	-3.6%
Questing	-17.9%	-16.5%	-24.5%	-25.6%	0.3%	-3.4%

We evaluated message aggregation as a mitigation strategy. Our default simulation transmitted each message as soon as it was entered into the simulation, incurring protocol overheads for every message, such as IPv4 and TCP headers. Given the small size of transmitted message payloads – 36 bytes on average [2] – this results in a message transmission overhead of more than 100% on average. We suspected that message aggregation, in addition to reducing bandwidth requirements, might have other benefits.

2) Message aggregation for latency reduction. Latency is an important metric for online game experience quality. Previous work suggests game messages should be sent as soon as they are available to minimize latency. Our simulations found the opposite, that message aggregation actually lowered latency.

Our simulator models TCP windowing, but not common TCP packet aggregation strategies. Instead, we implemented a simple time-based aggregation with a fixed aggregation window. We believe these results will apply to TCP with appropriate aggregation algorithms, and to 'custom' TCP-like IP protocols as proposed elsewhere.

Unfortunately, our simulation trace generation algorithm has some inaccuracy in assigning message initiation times. Simulation traces were created by instrumenting a small number of clients (one or two), recording client-server data streams, and extrapolating message transmission times. Avatar movement messages have game timestamps which allows us to assign their initiation time correctly. No other message type has these timestamps, so we relied upon IP packet capture time. During busy intervals we received large numbers of messages aggregated by the server. These aggregated messages are assigned identical or very similar event times in the simulation trace. These events should be more uniformly distributed, and so our analysis is biased towards showing benefits of aggregation which may be less prominent in more accurate traces.

In our case, aggregating messages and transmitting aggregated messages once every millisecond resulted in significant reduction in bandwidth consumption, in peak latency, and in some cases in average latency. We found this to be true for both the client-server case and the P2P Pub-sub case.

Earlier we presented statistics for non-aggregated client-server and pub-sub simulations. Below are two more tables showing the percentage change in attributes when aggregation is introduced.

Table 6 shows the changes to client-server traffic and latency attributes when all parties aggregate messages in 1 ms windows before transmitting them. This strategy introduces an average of 1 ms latency (0.5 client to server, and another 0.5 server to destination nodes) in end-to-end message delivery from aggregation delay. However, the aggregation allows more efficient transmission of messages from client to the server, reducing total bytes transferred. Even this small aggregation interval significantly reduces both average and typical peak bandwidth consumption.

In the client server case there is a small increase in average latency (less than a percent), but typical peak latencies are reduced by several percent, meaning performance during very active intervals is improved.

Table 7 shows P2P pub-sub improvements across the board. The most dramatic improvements are seen in average and peak latency. Across all scenarios, average latency is reduced by nearly 42 percent, halving it. For the highest latency scenario – raiding – average latency drops to one fourth of its non-aggregated value, and typical peak latency is reduced to less than half its original value. While the results would still provide a poor experience during active intervals, it would be dramatically better than the non-aggregated case.

Since aggregation provides benefits in both resource-rich (client-server) and resource constrained scenarios, we believe it should be an integral part of any messaging strategy for games which are typified by small but frequent message exchanges. This approach will ease network and node burdens, and provide a better end-user experience.

We did not investigate region servers in this work, although they are the middle part of the spectrum between client-server and pure P2P solutions. If practical concerns can be overcome, a region server approach is more likely to be deployable than a P2P solution.

V. Conclusions / Future Work

Massively Multiplayer online games (MMOG's) are a popular form of distributed virtual environment. A significant body of work done over the past ten years studies the behavior of MMOG's, and proposes ways to enhance scalability and performance. However, much of this work is evaluated against idealized workloads, vastly differing from researcher to researcher, making comparison difficult.

We captured a variety of interactions within World of Warcraft (WoW), and created simulation traces from messages issued by the game, rather than a logically deduced workload. This allows us to create simulation traces consistent with a broadly deployed game, and simulate projected network behavior for today's games. To help our evaluation's accuracy, we found and applied typical residential broadband bandwidth characteristics.

Table 7 - P2P Pub-Sub, 1 ms aggregation changes

	Up (kbps)	Up Peak (kbps)	Down (kbps)	Down Peak (kbps)	Avg Lat. (S)	Peak Lat. (S)
Capital	-6.5%	-0.7%	-7.1%	-7.2%	-22.3%	-20.7%
Instance	-28.8%	-19.4%	-29.0%	-18.0%	-60.2%	-55.5%
PVP	-17.5%	-8.0%	-18.6%	-22.5%	-52.5%	-42.5%
Quest	-10.2%	-8.3%	-13.6%	-11.7%	2.2%	-3.6%
Raid	-34.3%	-11.6%	-35.7%	-12.8%	-76.9%	-59.4%
Average	-19.5%	-9.6%	-20.8%	-14.5%	-41.9%	-36.4%

Finally, we implemented a packet-based network simulator using those bandwidth models, and modeling key network attributes such as inter-node latency, and TCP windowing. We compared bandwidth consumption and end-user latency of player nodes in client-server and P2P pub-sub message propagation schemes.

We found client-server topology did a good job of delivering messages to clients quickly and with reasonable bandwidth consumption. P2P pub-sub was unable to deliver messages in a timely fashion, often saturating client links, despite several favorable simplifying assumptions in our simulation. We conclude that pure P2P pub-sub is not feasible for MMOG's such as World of Warcraft and other similar games with current residential broadband.

We investigated message aggregation for bandwidth reduction. We found to our surprise that not only did message aggregation reduce bandwidth consumption, it improved message latency in most scenarios, and provided significant latency improvements for P2P pub-sub solutions. For future work, we believe it is worth investigating region server approaches using realistic traces, including leavers and joiners, node reachability, and bandwidth limits. We believe these factors will strongly influence evaluation of appropriateness of region servers. Many good region server schemes have been proposed in the past, and would be a good starting point for such research.

VI. Bibliography

1 Woodcock, Bruce Sterling. MMOG Subscriptions Market Share - April 2008. *MMOGChart.com*, http://www.mmogchart.com/Chart7.html (Apr. 2008).

2 Suznjevic, Mirko, Dobrijevic, Ognjen, and Matijasevic, Maja. MMORPG Player actions: Network performance, session patterns and latency requirements analysis. *Multimedia Tools Appl.*, 45, 1-3 (2009), 191-214.

3 Pittman, Daniel and GauthierDickey, Chris. A Measurement Study of Virtual Populations in Massively Multiplayer Online Games. (2007), ACM Press, 25-30.

4 Miller, John L. and Crowcroft, Jon. Avatar Movement in World of Warcraft Battlegrounds. In *Network and System Support for Games* (2009), IEEE.

5 Krause, Stephan. A Case for Mutual Notification: A survey of P2P protocols for Massively Multiplayer Online Games. In *NetGames '08 Proc.* (2008), ACM Press.

6 Lui, John C.S. and Chan, M. F. An efficient partitioning algorithm for distributed virtual environment systems. *Parallel and Dist.Systems, IEEE Trans. on*, 13, 3 (2002), 193-211.

7 Matsumoto, Nobutaka, Kawahara, Yoshihiro, Morikawa, Hiroyuki, and Aoyama, Tomonori. A scalable and low delay communication scheme for networked virtual environments. In *IEEE GlobCom Workshop 2004* (2004), 529-535.

8 Morillo, Pedro, Orduna, Juan M., Fernandez, Marcos, and Duato, Jose. Improving the Performance of Distributed Virtual Environment Systems. *Parallel and Dist. Systems, IEEE Trans. on*, 16, 7 (2005), 637-649.

9 Rueda, Silvia, Morillo, Pedro, and Orduna, Juan M. A Saturation Avoidance Technique for Peer-to-Peer Distributed Virtual Environments. In *Cyberworlds '07 Proc.* (2007), IEEE Computer Society, 171-178.

10 OfCom. UK Broadband Speeds 2009 (2009), 113.

11 Svoboda, Philipp, Karner, Wolfgang, and Rupp, Markus. Traffic Analysis and Modeling for World of Warcraft. (2007), 1612-1617.

12 Pang, Jeffrey, Uyeda, Frank, and Lorch, Jacob R. Scaling Peer-to-Peer Games in Low-Bandwidth Environments. In *IPTPS '07 Proc.* (Bellevue 2007).