Aspects of Networking in Multiplayer Computer Games

The authors promise to overview the following four aspects of MCGs: 1) Networking resources and how they provide the boundaries in which the MCG must operate. 2) Distributed networking architectures, including peer-to-peer, client/server, and server/network as well as control architectures, centralized, distributed, and replicated. 3) Scalability and parameterization. 4) Security against cheating, and gambling. This seems a reasonable division of the important concepts of MCGs.

I like the division of "shared space technologies" in Figure 1. Artificiality is plotted against Transportation. Artificiality being the degree to which a space is computer generated and transportation indicating how far away from their local space that a user is transported by the MCG technology. This is an interesting graph. I have seen this division of space done before in other fields and I think that once it is done, other works tend to follow along and fit into one of the spaces. This is a good example for our current work which will also provide a graphical division of research space. The authors spend some time explaining each of the four divisions of the research space. That physical reality takes place in local space, virtual reality takes place in a simulated world, telepresense in a remote area with real world objects, and augmented reality in local space with synthetic objects overlaid upon real world objects.

Three branches of research within the virtual reality space are identified: 1) Military simulations, 2) Virtual reality (hmmm), 3) Computer supported collaboration. The related work is identified (I assume that this covers the state of the art for 2002), particularly the DoD's Distributed Interactive Simulation, and High Level Architecture, which are both focused on large scale systems. Several examples of small scale systems research into Distributed Virtual Environments are given, and additional examples of Collaborative Virtual Environments are given which focus on the interaction of avatars. The authors note that all of these efforts are relatively unknown to the MCG industries designers and that this work is intended to address this problem.

I think that the abstract and introduction to this paper are very well written. The authors have stated 4 goals of the paper (listed above), divided the research space into 4 areas, chosen an area, and provided a motivation for the work. I am satisfied at this point that if the authors follow through on what they have promised (I can see from the section headings that they have) that the rest of the paper will be a slam dunk.

Minor edit #1 – Grammatical error in page 2, column 2, paragraph 1: Unicast communication between a single sender and a single receiver allows to control and direct the traffic.

Minor edit #2 – Grammatical error in page 2, column 2, paragraph 1: messages are intended to multiple receivers

The authors discuss network resources as a limitation that MCG applications must be designed to live with. They discuss bandwidth, latency, and computational power. This is a reasonable representation of network characteristics. The bandwidth discussion is rather limited. Covering LAN broadcast, WAN unicast, and suggesting multicast as a solution to limited bandwidth conditions in the WAN. I would have liked to see some discussion of the time variant aspects of bandwidth, and the difficulties in determining available capacity at the bottleneck router.

Minor edit #3 – Page 2, column 2, paragraph2: length of time that incurs when a message -> length of time that is incurred when a message

Latency and jitter are discussed next. The fact that latency cannot be completely eliminated because of speed of light, cable slowdown, and routing delays is discussed, and some examples of typical delays are given, e.g. 80 ms trans Atlantic RTT. An acceptable range of latency from 0.1 to 1.0 seconds is given. I would like to have seen a reference for this but it seems reasonable. The DIS specification of 200ms latency is mentioned. The authors discuss the differing requirements of applications, for instance a real time strategy game can tolerate more delay than a first person shooter as long as the jitter is low.

Finally the authors deal with computational power as a limitation. I think that this is still a problem from the point of view that more interaction requires more processing, i.e. the application must decide which packets to send to which sockets. However, I think that the problem relative to modern network cards is less severe. Today's NCs are much more capable than those of 2002, and most contain at least some processing power of their own. Once the packets are placed in the sk buffer the host processor no longer needs to worry about them.

In section 3 the authors deal with distributed architectures. They make the case that the resource limitations described above cannot easily be changed and therefore we should attempt to relax the requirements by changing the architecture. A single node is discussed briefly, where a split screen is used to provide multiplayer capability and all communication is internal to the node. Next p2p systems are discussed. The p2p systems considered in this paper are flat architectures where every node sends every message to every other node. This certainly lacks scalability, but there are hierarchical p2p systems available these days. Use of hierarchy in p2p would certainly help to address the scalability issue. Server centric systems are discussed where the server is the communication bottleneck and all messages pass through it. Finally the authors discuss server/network systems where a pool of well connected servers act in a p2p fashion while serving a network of connected clients. This has obvious tradeoffs in complexity vs. scalability.

Next the authors discuss data and control architectures. The tradeoff between consistency and responsiveness brought to light in that high responsiveness requires that computation be moved into the nodes while high consistency requires that the nodes query the data more frequently. Two types of relay into the network are discussed. A two-way relay that simply sends data into the network and then receives data back offers the highest consistency but the least responsiveness. A short-circuiting relay does the same but in addition it sends the data back into the local interface.
Minor edit #4 – Grammatical error: Page 4, column 1, paragraph 3: Basically, it is shared database -> Basically, its shared database.

Three different architectures are discussed, centralized, distributed, and replicated. In the centralized architecture one node acts as the shared database for all other nodes. The authors state that all other nodes must use a two-way relay in this case for consistency requirements. It seems to me that if the consistency requirements were relaxed there is no reason why short-circuiting relays of some form could not be used here.

Minor edit #5 – Distributed and replicated architectures suit better for -> are better suited for.

The authors provide an interesting observation that indeterminism leads to distribution and that determinism leads to replication. This is supported by the fact that for player controlled entities whom have only one source of command input and whose actions are indeterminate distributed architecture is best. Each character entity need only know about it's portion of the world. While for non player

characters replication is best because each NPC must take the same actions relative to all players.

In section IIIC the authors look at compensation techniques designed to reduce the load requirements on network resources. These are further divided into three techniques; 1) Message compression and aggregation. 2) Interest management. 3) Dead reckoning. Compression techniques save network bandwidth but require CPU resources to decompress. The authors say that aggregation saves bandwidth (certainly true) however, I am not sure it is clear how aggregation would affect responsiveness. I would have liked further discussion or an example. Certainly if a node had to wait for enough messages to accumulate to meet some aggregation threshold this would hurt responsiveness but I think that an aggregation technique that aggregates when messages are queued for aggregation and just sends un-aggregated traffic on when there are no other messages waiting would respond more quickly. I think there are tradeoffs to be researched here.

Interest management can certainly reduce the burden placed on the network by sharing only data where a players foci intersects another players nimbi. However, this will increase control traffic. The tradeoff between control traffic and reduced network load was not discussed. For instance, if all players nimbi are intersecting with each others foci then all data must be shared and the control traffic required to calculate this places additional burden on the network. However, in most cases a reduction of network burden should be realizable.

Minor edit #5 – Grammatical error, page 4, column 2, section c.3: which allows to prolong the interval -> which allows prolongation of the interval.

Dead reckoning is the technique where an object is updated infrequently but based on history or velocity and direction data its position is interpolated between updates. The authors say that the frequency of updates can be determined by an error threshold. This needs a little more discussion. If I have calculated the position by dead reckoning as accurately as possible how would I know how much error exists in the system until the next update? Are the authors referring to a history of errors? What are the tradeoffs here?

In section four the authors discuss scalability. They divide this topic into two sections, serial vs. parallel execution, and communication capacity.

The authors do a brief review of Amdahl's law and discuss three different levels of interactivity. 1) Separate real time games run in parallel (fully parallelized, no interactivity). 2) A turn based game. (Fully serialized and interactive but not real time unless the turns are very short). 3) An interactive real time game that has both serialized and parallelized components. They do some “back of the envelope” calculations giving an upper bound on the number of clients that can be serialized by 1 server.

Having calculated an upper bound for communication the authors produce a table demonstrating the capacity requirements (lower and upper boundaries) for each architecture. The upper bound in clients supportable by one server demonstrates that sublinear capacity requirement is needed. The hierarchical system best supports this, although a little space could have been devoted to discussing why this is true. The authors suggest using interest management, as well as compression and aggregation techniques to achieve this sub-linear requirement.

Finally in section 5 the authors discuss security and cheating. The authors state that the issue has not been addressed in scientific literature. This is clearly no longer true. Two goals of cheating are defined (other than hacking credit card numbers which falls into another class of malicious behavior, i.e.

stealing). 1) Vandalism, some malignant individuals will simply create havoc for the sake of destroying the game for others. 2) Dominance, some cheaters will wish to build super characters with powers they would not have otherwise been able to obtain. Three methods of cheating are examined: 1) Packet and traffic tampering. 2) Information exposure. 3) Design defects. Traffic tampering techniques discussed are as follows. Reflex augmentation, this is where a player uses software such as an aimbot. The aimbot tracks movement of other players and predicts there position when firing at them. Techniques using a proxy are discussed, such as intercepting damage packets and preventing them from reaching a user. A significant amount of space is used to discuss replay attacks. These are easily defeated with sequence or pseudo random numbering. This space would have been better used for the additional discussion that I noted above. The authors discuss information exposure in the context of compromised client software. They suggest having the servers check on client actions and take punitive actions against cheaters. Finally the authors briefly discuss design defects such as running trusted clients. They mention that this problem can be mitigated by binary checking but state that it is better addressed by by designing out the trusted clients. I tend to agree with this statement. In addition, they mention that some design factors may cause unexpected behavior under high latency or DOS attacks.

The conclusion is simply a wrap up of what they did in the paper and a few possibilities of future work tacked on the end. I suppose this is fair since this is an overview paper. I think that the paper is a little light on scientific contribution, but being a survey I suppose that is okay. Also I am not familiar with the state of the art in gaming as of 2002. Perhaps the contributions are of more weight when viewed from the 2002 perspective. I think that the paper is very well written in spite of the grammatical mistakes (I stopped correcting them after 5 or so errors). The authors clearly know what they are doing when it comes to layout and flow of a paper. The goals of the paper were clearly laid out in the abstract, motivation was provided in the introduction, and the body of the paper went on to accomplish each goal stated in the abstract. The conclusion, stated each goal that was accomplished in the body of the paper and suggested future work. Very nice, very formulaic. That being said I did not feel intellectually satisfied with the depth of information provided on each topic. Perhaps that is because I am looking back from a perspective of 8 years later, perhaps it was done to meet the page requirements. Still I don't feel that I have been completely enlightened after this survey. All of the components are there, but none of them are outstanding. I give it a weak accept.