Paper Analysis

Aspects of networking in Multiplayer Computer Games

Published in 2002, this paper comes at the time when multiplayer games had started to make significant inroads into households, that is when the number of users in such games had started rising. Thus the paper provides a brief but important overview of the challenges that such games would present to the networks via which they would be played.

The authors start off by describing what 'shared space technologies' are. A concise but very informational graph is provided which clearly distinguishes the various classes of these technologies present. I particularly like these graphs as they epitomize the phrase 'a picture speaks louder than a thousand words'! A similar graph was present in the previous (VoD) paper and was equally helpful. The four kind of technologies described are:

1. Augmented reality
2. Physical reality
3. Virtual reality
4. Tele-presence

The introduction section of the paper is devoted to the historical background of these technologies and several acronyms before coming to the point where the authors describe that the purpose of the paper is to plug the gap between the academic research and the challenges faced by these technologies. Thus they pick up the area of networking for finding the challenges.

Multiplayer games are basically distributed systems and they face basically three challenges related to network resources:

1. network bandwidth
2. network latency
3. host processing power for handling the network traffic

It was pointed out that the network bandwidth required is dependent upon several factors: frequency of messages sent, size of messages, number and distribution of users and the type of transmission used, that is multicast, unicast or broadcast. Next, the aspect of latency has been analyzed and is obvious, the strategy games are more tolerant to these delays than the first person shooter games (maximum latency recommended: 100ms).

The authors state that the problems of the distributed interactive systems are best tackled at the levels of choosing architectures for communication, data and control, which indirectly address the first two challenges listed above. I think this is where the real 'meat' (which is anyway pretty less) of the paper starts. The authors first describe the various type of architectures possible based on the degree of deployment: peer to peer (every node connected to every other node in the network), client/server architecture (one node is made a server and others act as client) and server/network architecture (multiple interconnected servers interact with the clients). These different types of architectures basically represent different extent of trade-off between the network bandwidth and latency. While client/server networks offer less latency, the network bandwidth required is very high near the server. The server-network architecture offers an interesting avenue to manage this trade-off in order to achieve the best results.

Next, the data and control architectures are taken up and are analyzed based on two aspects: consistency and responsiveness. The authors mention that to achieve high consistency,the processes should be tightly coupled, and to achieve high responsiveness, nodes should be loosely coupled. I think the authors should have explained this in a bit more detail. Though it is simpler to understand the tightly coupled issue aspect, the loosely coupled aspect is a somewhat unclear. A bit more explanation would have been quite helpful. The responsiveness of the system is described in terms of 'relays', I am not sure why they were used though when they could have just said that nodes need to send and receive control messages and data from a central server. A centralized architecture is good for maintaining data consistency (as the data is in one place), but is not terribly good for responsiveness. So, the authors argue that a distributed architecture is way more efficient (which sounds quite logical!).

Then the authors mention that these architectures are helpful only to a certain extent and in order to have a better performance, some compensatory techniques to reduce the communication between the nodes are required. The

techniques discussed are:

1. Message Compression and aggregation
   - The communication messages between the nodes are compressed and several messages are sent bundled together, thereby reducing the amount of bandwidth needed
2. Interest management
   - This is a fairly interesting idea which says that the nodes need to know only the information they have to know (or they are interested in). Hence, only that information is communicated to them. The authors did a fairly good job of explaining the concept with the help of figures and the examples of focus and nimbus.
3. Dead reckoning
   - This is an idea quite analogous to the I-B-P frames of video coding. The nodes locally compute the data based on some delta information and thus they do not need to regularly fetch this data from the server. However, the authors say here that messages need not be communicated regularly and need to be sent across only if the computation is very erroneous. My question is as to how the local node will know that its computation is erroneous if it is not receiving data? It would require some additional processing and information to be communicated. Maybe a mention of an idea how this could be done would have been a good thing.

Then the authors take up the issue of Scalability ("the ability to adapt to resource changes") and state something which confused me: To achieve scalability, there must be a physical parallelism that enables logical concurrency of computation. Were the authors referring to multiple servers? But all collaborating to perform one function? A more clear explanation would be great here. Scalability is based on some factors:
Serial and parallel execution: In this section, the authors size up some parameters to describe parallelism, specifically the 'speedup gained' factor. However, I found this a bit ambiguous (maybe only for me!). While explaining the Amdahl's law, the authors say: let $T_s + T_p = 1$ and a (alpha) = $T_s/(T_s + T_p)$. Now if the sum is 1, wouldn't the value of alpha be T_s ? However, the rest of the argument based on alpha, flows smooth as authors conclude that MCGs need both parallel and serial computation. Then, while concretizing to a value for the number of clients, the authors mention that each packet takes a frame size of at least 68.8+26.8 bytes. I am not very clear how that came about. Is it something to do with the IPv6?
Communication capacity: The authors here have computed the worst case bandwidth requirements for the architectures discussed so far and concluded that the scalability is achieved when sublinear communication is achieved. By sublinear communication, the authors mean that a client is not aware of all other clients all the time. Thus it also encompasses the interest management policies discussed earlier.

The last aspect the authors look at is 'Security and cheating' in the system. Security refers to the security of private sensitive data of the users. The common methods of cheating described here are one of the most interesting parts of the paper. Reflex augmentation, packet interference, packet replay are the techniques mentioned and they look fairly easy to tackle using cryptography. A few more interesting tit bits are mentioned here w.r.t. cryptography which though do not look like very important from the perspective of the whole paper. In fact, the rest of the topic deals more with security and cheating issues created by bugs in the server softwares rather than the network architecture.

Concluding, I think the paper was a bit lacking on thoroughness (my decision could be influenced by the length of the paper!) and perhaps it was only aimed at giving an overview instead of getting into the details of every aspect. Nevertheless, the authors presented the issues in a nutshell and offered very high level solutions to them.