1. Background: Large-scale multi-peer networks

Most of what we think about when we discuss online services and interactions today generally can be reduced to a series of simple one-to-one interactions between clients and servers. Even massive social networks on the scale of Facebook or Twitter that facilitate huge amounts of many-to-many interactions between individual peers follow this model; one-to-one client-server interactions effectively give the appearance of direct interactions between peers.

Problems arise, however, when the interactions between peers are both time- and bandwidth- sensitive as well as high volume (that is, a single exchange between a group of peers requires a large data transfer not necessarily a high volume of exchanges).

It is easy to envision a few key interactions between groups of individuals that meet these requirements. Sharing large files and streaming video and audio are obvious examples. Online audio and audio-video conferences clearly meet these requirements. The sheer volume of data transferred between peers in a single interaction is large, it is time-sensitive because one peer experiencing a lag in audio can ruin the experience, and it is bandwidth-sensitive, because if the nature of network traffic changes or some peers have a lower bandwidth connection than others, quality is negatively effected. However, the most salient example of modern day multi-peer online interactions whose scale is already outpacing many networks’ capabilities is online multiplayer gaming.

In a single online gaming session, each end user must be consistently updated in real time on the activities of every other end user as well as other changes to the game environment and they must also consistently upload their own activities so their peers can be updated. In fast-paced action games, this is clearly a heavily time-sensitive situation. Similarly a single user without the required bandwidth capabilities can be left with a poor user experience and negatively impact gameplay for other users. Because of this, without extremely well designed, dedicated servers that few entities have both the ability and incentive to maintain, online multi-peer interactions must be limited to relatively small scales.

A key problem in large-scale multi-peer online interactions is that internet connectivity is far from uniform across peers. This presents an issue to the network setup required for facilitating such interactions, which is that the limits of the interaction are not necessarily known from the outset.

whether or not the limitations of the connections of various peers are known, as we add more peers to the network, upload capacities increase linearly, but the number of updates required between peers to maintain synchronicity increases quadratically because each peer needs to update every other peer.

Some potential ways to alleviate this problem might be to design the network applications to only update interested parties. For example, the application could require each player to only update nearby players in an online game. However, this solution is only partial and breaks down when all or many parties are interested in updates, such as when audio or video conferencing is involved. Additionally, to alleviate the problem the network application might make higher demands of peers with strong connections, for example by asking them to forward data that weaker peers cannot keep up with. The “Donnybrook” system is an example of an attempt by Microsoft which presents yet another potential solution to this problem. It attempts to estimate what aspects of online gaming interactions are most important to players and convey those with high fidelity and simulate the rest with occasional, lower fidelity updates.

While there is great focus on this problem in gaming and other sectors for networked personal computers and gaming consoles, a rapidly growing portion of the networked community that will require more and more of these large-scale interactions with many peers is that of mobile devices and wireless networks. In the coming years, it will be necessary for mobile networks to confront the problems of large-scale time- and bandwidth-sensitive online interactions. For this reason, this project will focus on network interactions between mobile devices.

2. Description of Proposed Project

The goal of this project will be to develop a method of setting up a scalable network connection structure that facilitates large-scale, multi-peer interactions for mobile devices that are both time- and bandwidth-sensitive. The project can roughly be divided into three parts as follows:

1. The network application:
   The user-end application must have strong situational awareness regarding the state of the network between peers and servers and of the capabilities of both the peer and server. The application must be able to detect the upload and download capacities of the servers, it must be able to identify other peers and their capacities as well. For the application and as part of this project, I must come up with a method for determining the limitations of the network by probing both the servers and peers running instances of the application. Potential avenues of exploration include RTT packet analysis and tracing packet hops to get a good sense of the limitations of the peers on the network.
2. Reserving network resources and bandwidth
   The second part of the project will involve determining how to set up the
   connection structure of these interactions. To set up the connection, the
   application should be aware of its minimum requirements and take into
   account the capabilities of the various peers as determined in part one.
   However, the key to this stage focuses on determining a method of
   guaranteeing a minimum bandwidth from the various ISPs that peers are
   connected to and reserving that space for the duration of the single
   interaction. This step may involve looking into structuring design
   suggestions that can allow ISPs to charge by reservation of high capacities
   rather than by monthly capacities and limits. However, this stage is still
   very vague and will require a lot of further research.

3. Utilizing other resources of the network
   The final stage of the project, time permitting, will focus on using
   resources of the overall network shared by peers to improve
   performance. Caching within ISPs to improve update times between
   peers is one possibility. Using peers with high capacities as reflectors to
   facilitate updates is another possibility.

3. Tentative Schedule

   A tentative schedule for the project is as follows:
   1. Understand the structural details of peer-to-peer networks and the
      ways that ISPs dedicate resources to network users. Develop an
      understanding of how to approach the tasks described above (~2
      weeks)
   2. Develop an iOS or Android application that both provides a time- and
      bandwidth-sensitive service to the end users and also is capable of
      establishing a multi-peer network and determining the capacities of
      its various edges (~4 weeks).
   3. Develop a method of setting up a reserved connection for the duration
      of the sensitive online interactions to allow those interactions to
      proceed with high fidelity. (~3 weeks).
   4. Explore other abilities of the network to improve performance as
      described in part 3 above (remaining time).

4. Preliminary Sources

   - Donnybrook: Enabling Large-Scale, High-Speed, Peer-to-Peer Games
   - Provisioning On-Line Games
   - End-to-end transmission control mechanisms for multiparty
     interactive applications on the internet