Part 1: Paper Understanding

1. The paper claimed that the median latency on public LTE networks for AR apps was about 12.5 seconds and that the radio access network accounts for about 30% of this latency. They then go on to claim that their solution increases network performance and improves the end-to-end AR latency and goodput by 40% - 70%.
2. AR is different from other multi media applications because it is much more sensitive to latency. With other MMA it is possible to start drawing or painting images or showing frames before the entire data set is received, this is due to the ability to paint information as it is received and to use a frame buffer to store future frames for videos. In AR the resolving user cannot see the object until the host user has sent its entire data transmission, this results in the potential for users to try to interact with objects that have already been moved or deleted.
3. In this paper the problem addressed is based on multi user AR over cellular networks. Other state of the art related works were based on single user AR object detection with cloud processing or SLAM based robotics with a focus on SLAM algorithms rather than communications.
4. The end-to-end latency for AR is broken down into the handshakes, visual data transmission, cloud processing and local rendering for the hosting device, which was found to take about 93% of the total end to end latency for public LTE networks, and data preprocessing, visual data transmission, cloud processing and local rendering for the resolving device, which was found to take about 7% of the end-to-end latency for public LTE networks. The visual data transmission for the host device is further broken down into the TCP/IP and radio access network components as it is large percentage of the latency.
5. The authors found that a very large amount of the data transmission happened on the uplink from the host device whereas the downlink of the resolving device was negligible, (2.5MB vs 100 KB). It was found that AR has traffic bursts when something happens time delay between these spikes is based on user input making it difficult to predict. These spikes create congestion in TCP as all of the data goes through the same TCP stream.
6. The authors propose choosing smart packets sizes for transmission based on the network congestion. They found that und heavy network congestion packet sizes of 650-bytes was the most efficient whereas when congestion was low the maximum packet size of 1430-bytes was acceptable.