**Measurement Study of AR/MR Platforms**

Authors: Shilpa Chetty Vobbilisetty, Reethu Gogineni

**Steps to use Wireshark**:

To ensure accurate measurements during a specific experiment, it is essential to connect only to required IP and avoid recording all the available networks’ data.

1. Connect your PC to Mason-Secure WiFi/any other available WiFi
2. On your PC, go to Settings → Network & Internet → Mobile Hotspot → On. This network is shared over WiFi.
3. Connect your HoloLens to this network.
4. Once connected, you will be able to see an IP address of HoloLens below. Note it down. Say it is 192.168.137.246
5. Open Wireshark application on the PC.
6. Under the capture filter, you can see various network options. Hover over them to see their IP address. Double-click on connect to the one which has your HoloLens IP - 192.168.137.xx
7. As soon as you connect, you should be able to see packets being transferred.
8. Once the experiment is done, you can stop capturing packets by clicking on the red button below ‘File’ option.
9. I/O graph can be checked from Statistics → I/O Graph
10. Display filter can be added to see relevant information

Example - [ip.src == 192.168.137.246]

[ip.dst == 192.168.137.246]

And the Y-axis should be selected as ‘bits’ from the drop-down. It is set to packets by default.

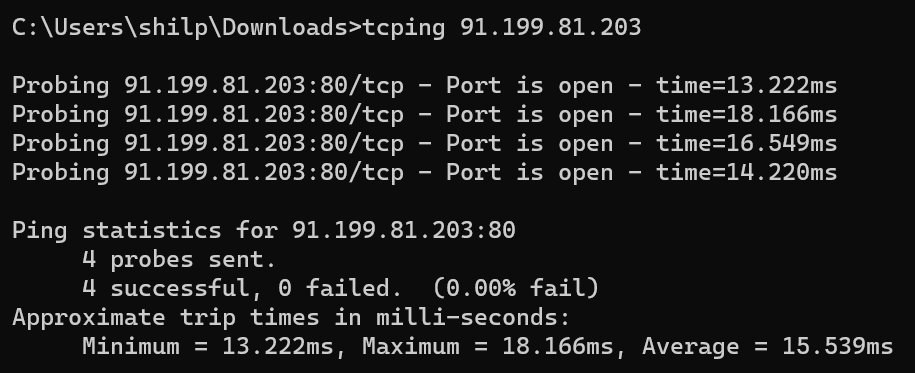
1. Next you can check for conversations that happened during the course of the experiment from Statistics → Conversations → IPv4. You can sort according to your desired fields for analysis.

**Steps to use tcping**:

1. Download executable from <https://www.elifulkerson.com/projects/tcping.php>.
2. Open command prompt, change directory to the location where you placed the above .exe file.
3. Next you can say tcping followed by the IP address you wish to find RTT for.

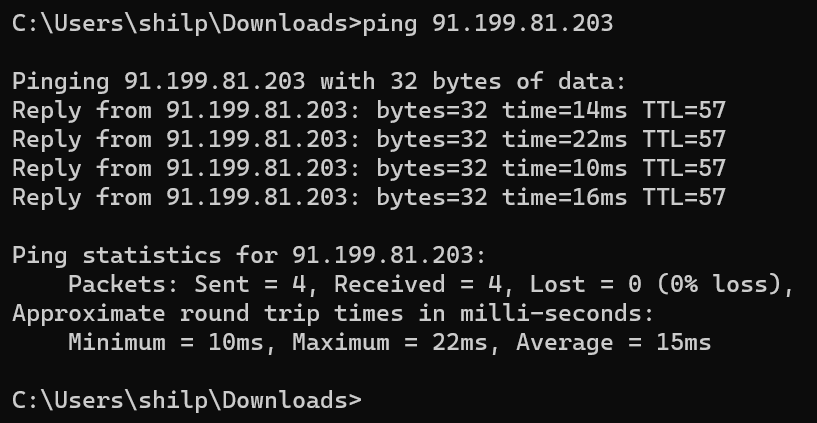
Example - tcping 91.199.81.203

1. The average value is considered as RTT.



**Steps to use ping:**

1. Open command prompt
2. type ping followed by the IP address you wish to find RTT for.
3. Example - ping 91.199.81.203
4. The average value is considered as RTT.



**Steps to use tracert:**

tracert (for windows) or traceroute (for Mac) can also be used for finding the RTT between two servers.

1. Open command prompt
2. type tracert followed by the IP address you wish to find RTT for.
3. Example - tracert 91.199.81.203
4. The RTT is the average of the elapsed time for each hop for the desired server.

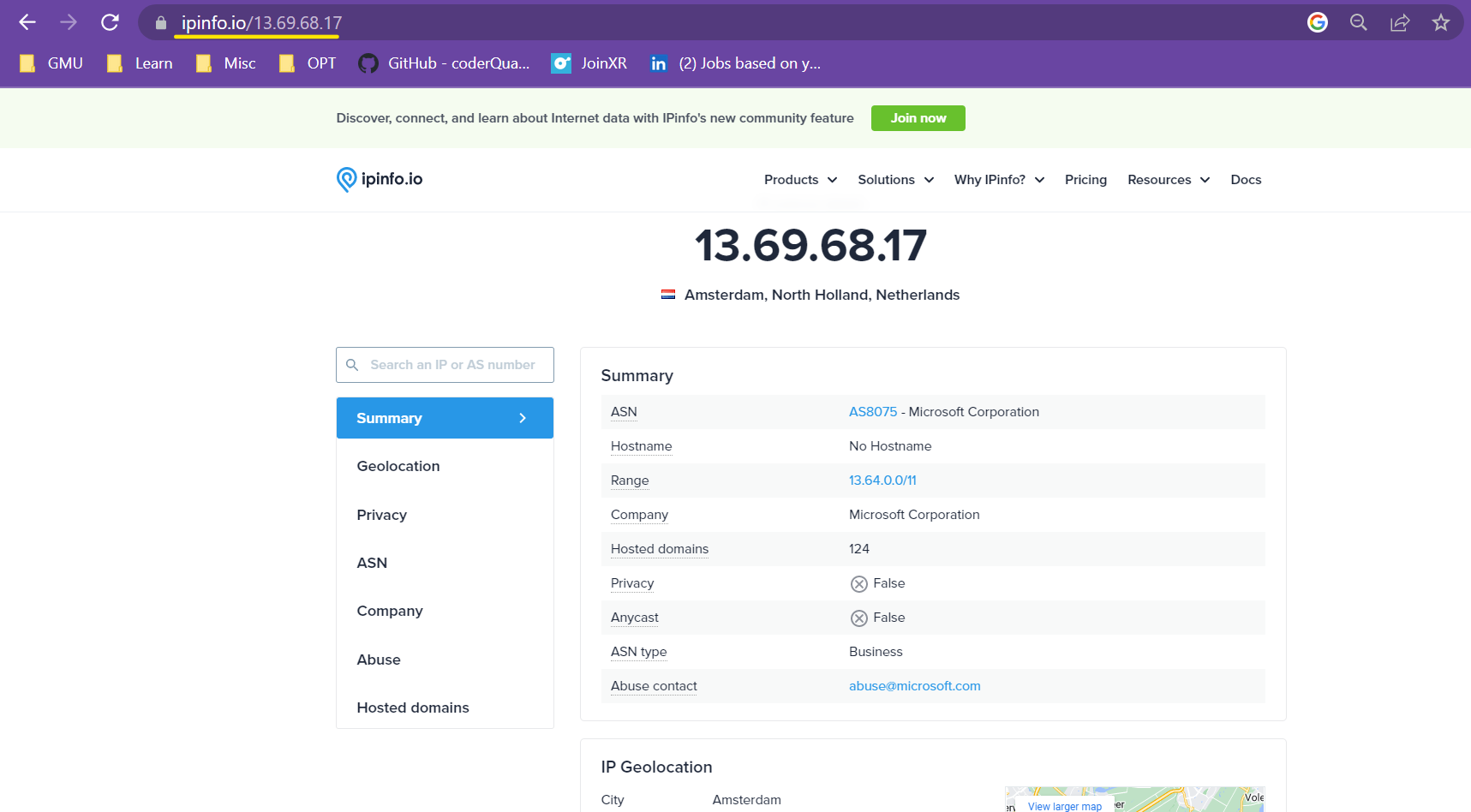


RTT = average (17, 17, 15) = 16.33 ms

We primarily used tcping. If it returned ‘no response’, it happened that even ping/tracert responded the same way.

**Steps to use ipinfo**:

1. To find the geolocation, anycast of any server, ipinfo can be used.
2. You can simply type ipinfo.io/<server IP> in your favourite browser
3. Example - ipinfo.io/13.69.68.17



**Steps to calculate (end-to-end) E2E latency**:

Ensure that there is a milli-second clock in front of both the users. [[clock](https://codepen.io/jasonleewilson/pen/gPrxwX)]

1. Start video recording in the hololens before starting the experiment.
2. Once the experiment is done, stop recording.
3. Save the video to your desired location on the PC.
4. Download ffmpeg from <https://ffmpeg.org/download.html>
5. Add it to path
   1. PC taskbar → search → edit the system environment variables → environment variables → path (under user variables) → edit → new →

<your location> \ffmpeg\bin → click ok

* 1. Under system variables → new,

variable name: ffmpeg

variable value: <your location> \ffmpeg\bin

1. Open command prompt → cd <video-location> →

type ffmpeg -i <video-name>.mp4 %04d.png and hit enter. The frames will be in the same directory as the video.

Number of frames = seconds in the video x 30 frames/sec

**Calculation**:

For example , in an experiment for visualising 0.1m using a measurement tool, User A observes a measurement of 0.1m at 12:05:31:180, while User B observes it at 12:05:31:750. The E2E latency is:

12:05:31:750 - 12:05:31:180 = 570 ms → eq 1

The result can be cross-verified with the frame difference method. Firstly, the difference between the target frame and the initial frame is calculated. Then it is converted into time difference by multiplying it with 33 ms (30 frames are generated every second, which means that each frame is generated in 0.33 seconds or 33 milliseconds). Finally, this value is added to the initial time. The resulting time should be similar/close to the time the users saw on the clock for 0.1m.

**User A**:

1. Initial frame number - find the one that has clear milli-second time.

example: 0005 and time is 12:05:16:500

1. Target frame - say, you are looking for 0.1m and it it first visible in the frame number 0450

**User B**:

1. Initial frame number - find the one that has clear milli-second time.

example: 0001 and time is 12:05:16:800

1. Target frame - say, you are looking for 0.1m and it it first visible in the frame number 0455

User A - Initial time + (target frame number - initial frame number)\*33 ms

- 12:05:16:500 + (0450 - 0005)\*33 ms

- 12:05:16:500 + 14685 ms

- 12:05:16:500 + 00:00:14:685

- 12:05:31:185 (close to time method: 12:05:31:180)

User B - reference time + (target frame number - reference frame number)\*33 ms

- 12:05:16:800 + (0455 - 0001)\*33 ms

- 12:05:16:800 + 14982 ms

- 12:05:16:800 + 00:00:14:982

- 12:05:31:782 (close to time method: 12:05:31:750)

***E2E Latency*** = 12:05:31:782 - 12:05:31:185 = 597 ms → eq 2

(or) other method with :

Since the reference time of user A is less compared to user B,

= 0

= 12:05:16:800 - 12:05:16:500 = 300 ms

User A = 0 + (0450 - 0005)\*33 ms = 14685 ms

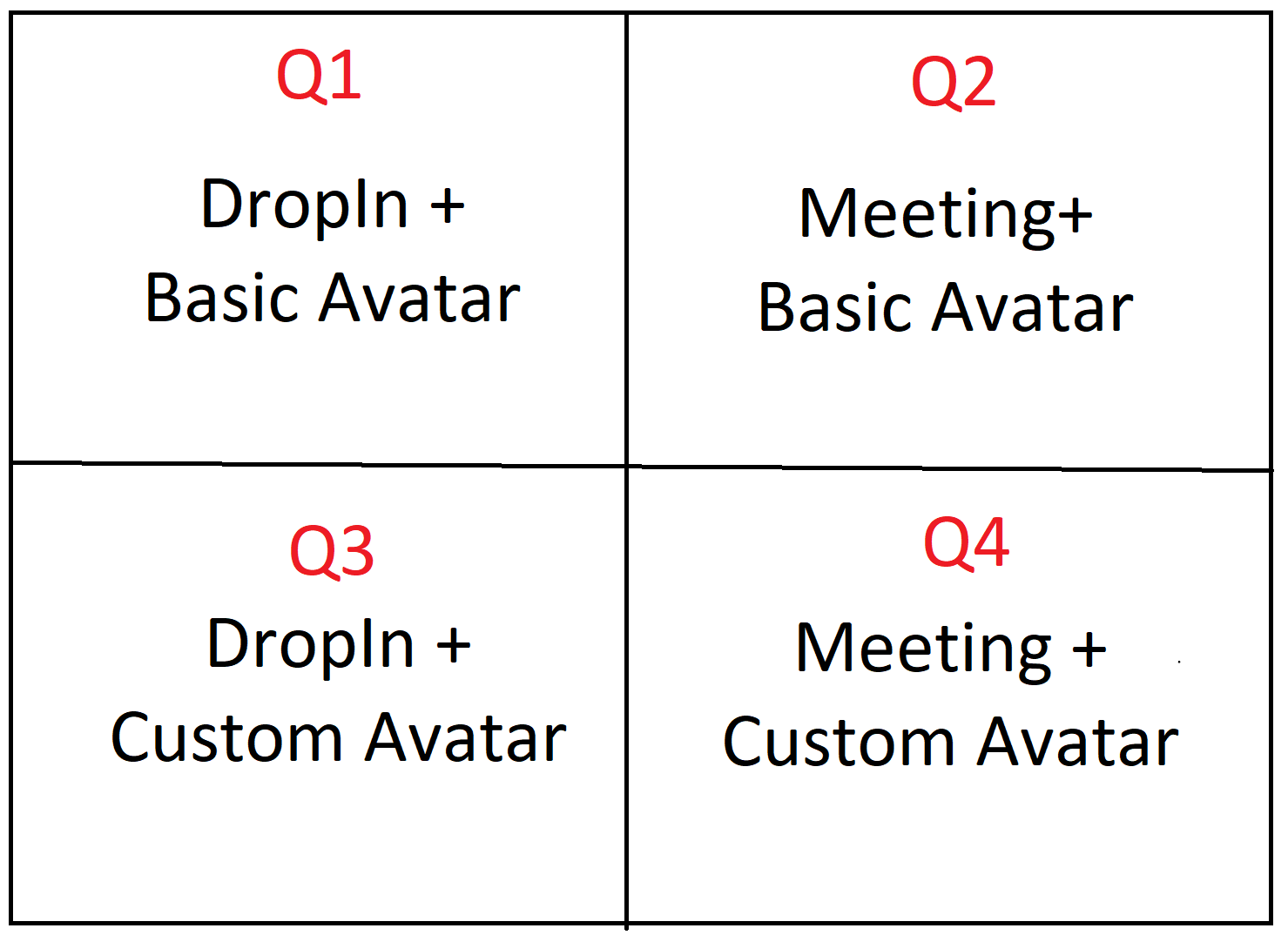
User B = 300 + (0455 - 0001)\*33 ms = 15282 ms

***E2E Latency*** = 597 ms → eq 3

It should be noted that the values for equation 1 and 2 are the same. And the values for equation 1 and 2 are close to each other (~ 22 ms). The frame difference method alone can also be used to calculate E2E latency.

**Our experimentation procedure:**

We calculated the E2E latency of a tool/action using joinXR in four different scenarios, mentioned below with four quadrants:



In the first quadrant (Q1), we used the DropIn method to join the meeting using the basic avatar(from fig a) which is the default avatar in JoinXR.

In the second quadrant (Q2), we created a meeting for a particular time slot and joined the session using the basic avatar(from fig a).

In the third quadrant (Q3), we used the DropIn method to join the meeting. However, we have customised our avatar (from fig b) by uploading our own image before joining the meeting.

In the fourth quadrant (Q4), we used the Meeting method to join the meeting with the customised avatar(from fig b).

Every experiment is performed 5 times in all the 4 quadrants. After calculating the individual latency, we calculated the mean and standard deviation for each quadrant.

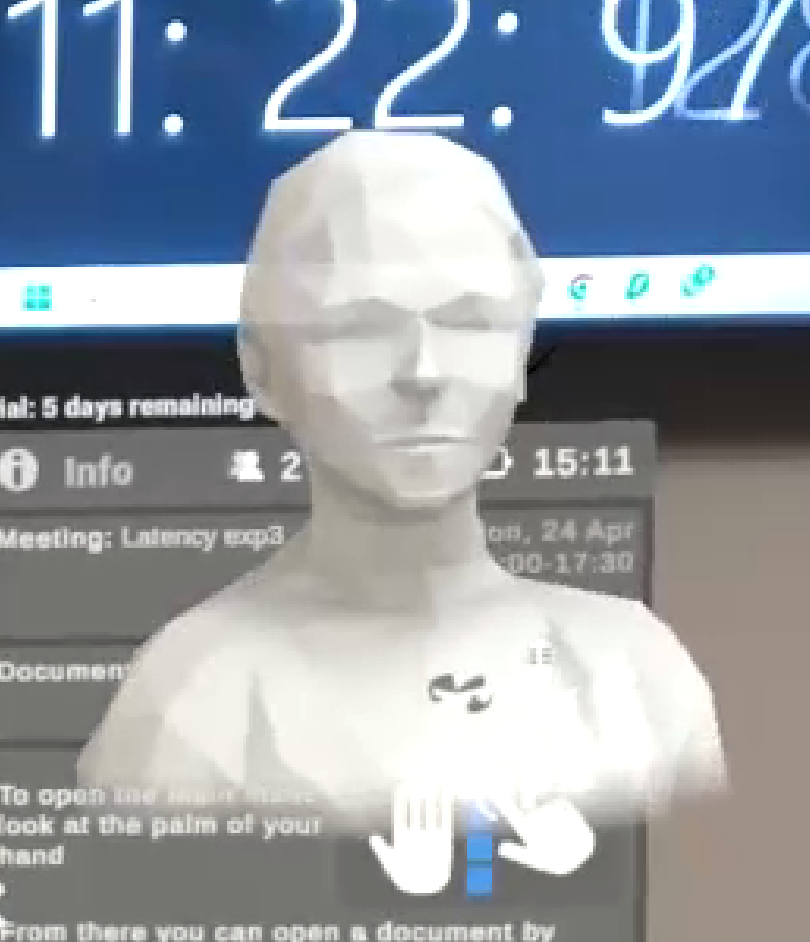
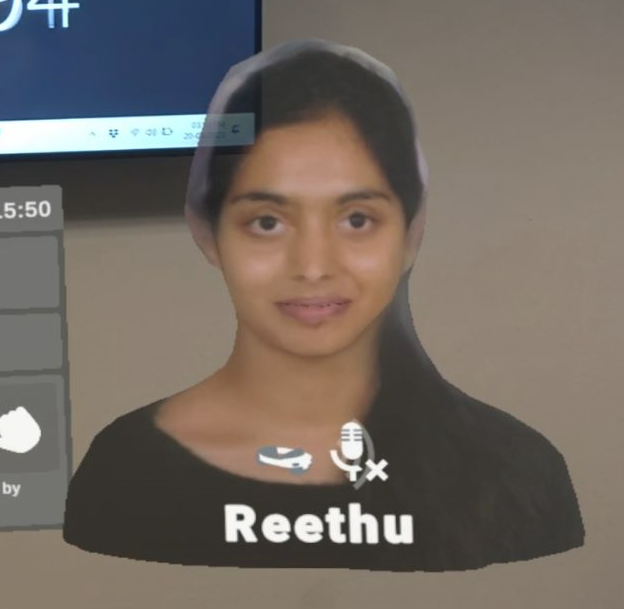
 

Fig a: Basic Avatar Fig b: Custom Avatar