

Supplementary Materials: Measurement of VCAs

1 TRAFFIC DECODING

1.1 Traffic Hierarchy

Through our packet tracing, we observe that UDP is employed for the transmission of audio, video, and screen-sharing streams, adhering to the principles of the TCP/IP model [1]. However, an unexpected challenge arose with Zoom's implementation of a custom transport protocol over UDP, which precludes straightforward decoding of RTP packets. To address this, further exploration is necessary.

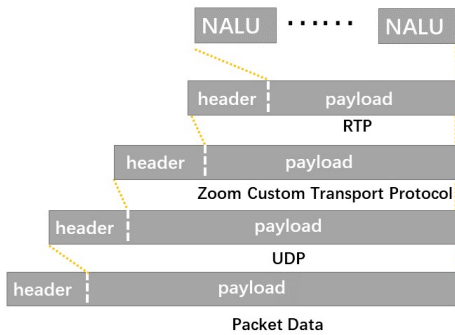


Figure 1: Structure of Zoom's Network Layer[2]

[2] provided significant insights into transport-layer analysis. They found that Zoom's UDP payload for audio, video, and screen-sharing flows begins with "\x50", providing a way to filter out control signals from the primary data streams. Moreover, each type of media stream in the UDP payload can be further differentiated: audio streams are marked by "\x0f0100," video by "\x100100," and screen-sharing by "\x1e0100." These unique identifiers allow for the effective separation and analysis of the three media sources.

1.2 Media source Classification

For Zoom and Webex, our packet analysis reveals that each platform utilizes three distinct local ports for the transmission of audio, video, and screen-sharing, respectively. This separation allows for the individual extraction and analysis of data streams for each media type.

In contrast, Google Meet employs a single local port for all transmissions, necessitating a different approach for distinguishing between media types. In this case, the "Payload Type (PT)" field within

the RTP header is instrumental. Specifically, a PT value of 111 indicates audio transmission, PT 98 denotes screen content, and PT 96 corresponds to video content.

Additionally, for Zoom, we identified unique PT values for each media type: 112 for audio, 98 for video, and 99 for screen-sharing. These distinctions facilitate precise identification and separation of media streams, enabling targeted analyses of transmission characteristics for each media type.

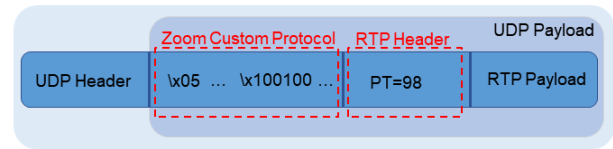


Figure 2: Example of decoding: decoding video flow to RTP

1.3 UDP & RTP Decoding

Using local port numbers and the "Payload Type" (PT) field, we effectively segregate and analyze each component of the media streams. Figure 2 illustrates a decoded structure of Zoom's video flow, tracing the path from UDP through Zoom's Transport Protocol to RTP.

The RTP packet header is crucial for detailed media data analysis. It contains several key fields: the "Sequence Number" is used to track each packet and detect any packet loss. The PT identifies the format of the media content. The SSRC field distinguishes between participants within a conferencing session. Lastly, the "Marker" field denotes the boundary of each video frame, facilitating precise frame-by-frame analysis. This structured approach allows for a comprehensive evaluation of transmission integrity and participant interaction during video conferences.

2 NETWORK UTILIZATION

Each video conferencing application (VCA) employs distinct strategies for managing multimedia transmission. To evaluate and compare their network characteristic, we conduct measurements using consistent audio and video inputs across all platforms without any network constraints. This approach ensures that any observed differences in performance metrics are attributable to the VCAs' unique processing and management techniques rather than variations in input data. Table ?? displays the network utilization for three media sources-audio, video, and screen - across various scenarios.

3 BANDWIDTH ALLOCATION ACROSS MEDIA SOURCES

3.1 Zoom

Figure 3 illustrates the traffic prioritization across four scenarios, consistent with observed patterns in downlink traffic. As bandwidth diminishes, the datarate of video and screen-sharing decreases, while the audio datarate remains relatively stable at approximately

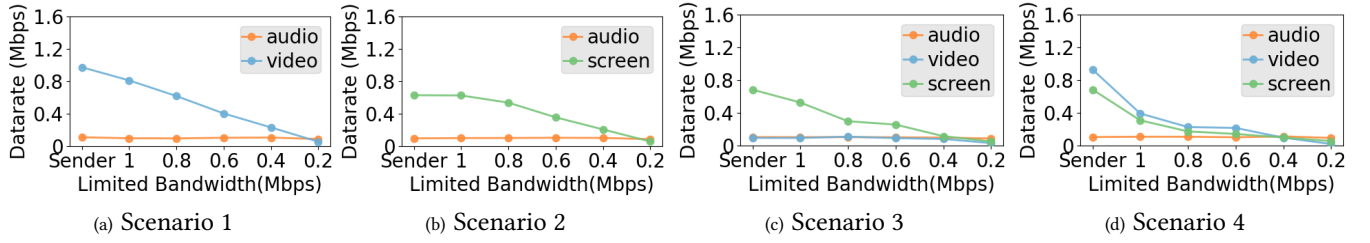


Figure 3: Zoom: average bandwidth under four scenarios with uplink bandwidth limits on each receiver

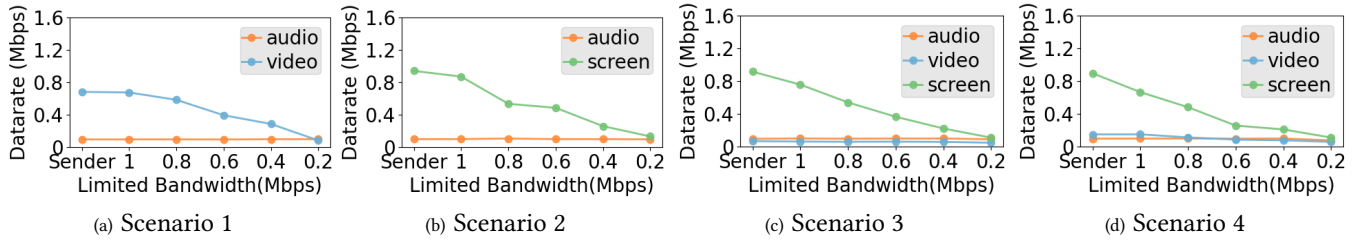


Figure 4: Webex: average bandwidth under four scenarios with downlink bandwidth limits on each receiver

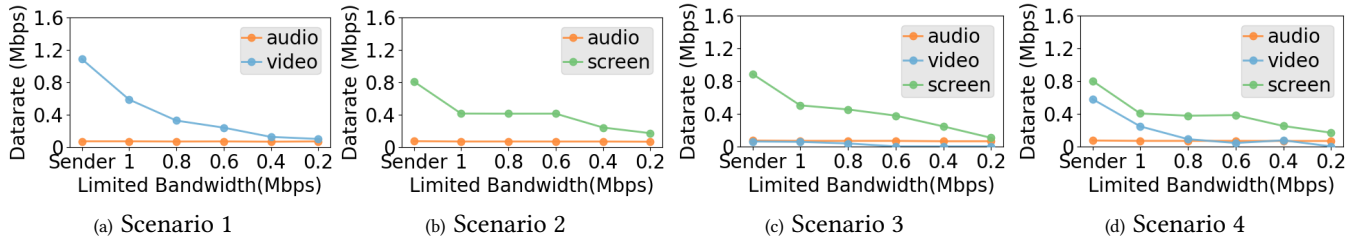


Figure 5: Google Meet: average bandwidth under four scenarios with downlink bandwidth limits on each receiver

	Audio		Video		Screen	
	send	receive	send	receive	send	receive
Zoom	100K	100K	1M	0.8M	1M	1M
Webex	795K	95K	700K	700K	1M	1M
Google Meet	70K	70K	1M	1M	800K	800K

Table 1: Datarate in multi-party video conferencing

100Kbps. Notably, even under severe bandwidth constraints (less than 400Kbps), the audio maintains a data rate around 100Kbps, the highest among the three media types, indicating a robust prioritization. In contrast, video consistently receives the lowest data rate at this time. These observations support the conclusion that Zoom prioritizes its traffic with a clear hierarchy: audio > screen > video.

3.2 Webex

Figure 5 demonstrates Webex’s approach to bandwidth allocation among three media sources. It is evident that the audio data rate remains consistent, while the data rates for video and screen-sharing

decrease as downlink capacity becomes more constrained. This pattern mirrors that observed in Zoom, where under significant bandwidth limitations, the audio data rate surpasses those of screen-sharing and video, with video data rates potentially approaching zero. This consistent behavior suggests that Webex employs a similar traffic prioritization strategy to Zoom, favoring audio over other media types. Similar trends can be observed when uplink bandwidth is limited, reinforcing this prioritization strategy.

3.3 Google Meet

Similar to Zoom and Webex, our analysis reveals that Google Meet prioritizes audio over video and screen-sharing in its traffic allocation. Additionally, in Scenario 4, we observe that the video data rate drops significantly even when bandwidth is not restricted too much (0.8Mbps). This behavior indicates a deliberate strategy by Google Meet to ensure the quality of screen-sharing is maintained, potentially at the expense of video quality.

4 CASE STUDY: ZOOM PACKET-LEVEL ANALYSIS

4.1 Video Transmission

we match packets between the sender and the five receivers to observe how Zoom manages video transmission. As illustrated in Fig.6, the sender encodes our 720p video input into three distinct resolutions—360p, 180p, and 144p—before uploading them to the server. The server then selectively dispatches these packets based on the downlink capacity of each receiver.

Receivers with higher data rates, such as Receiver1, Receiver2, and Receiver3, primarily receive the 360p stream, ensuring the best possible video quality under their network conditions. Conversely, in situations of limited bandwidth, these receivers experience packet loss, leading to a reduction in framerate. Meanwhile, Receiver4 and Receiver5, who have lower data rates, receive streams at 180p and 144p resolutions, respectively. This strategy ensures that the video remains relatively smooth with a guaranteed framerate, albeit at reduced clarity.

It is important to note that no receiver obtains all the packets initially sent by the sender. This selective packet dispatch explains the discrepancy between the video data rate at the sender side (1Mbps) and at the receiver side (0.8Mbps), illustrating the server’s strategic adaptation to varying network capacities across different receivers.

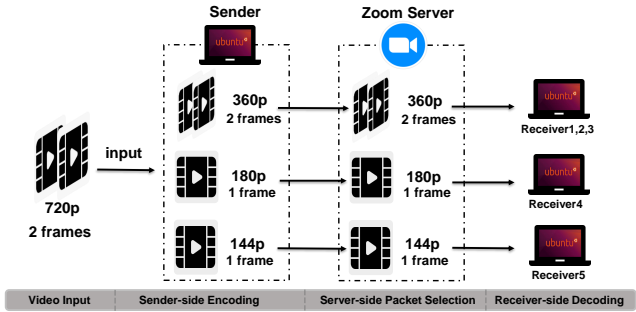


Figure 6: Three-resolution video transmission over different download constraints

4.2 Screen Transmission

In screen-sharing transmission, there is no resolution degradation mechanism employed to maintain the framerate; the resolution remains constant unless the content shared from the sender’s screen changes. Our analysis, through decoding UDP and RTP packets, reveals that packets corresponding to the same frame are generated simultaneously but are not always transmitted in sequential order. Moreover, packet loss does not typically occur within a single frame.

REFERENCES

[1] Mohammed M Alani. 2014. Guide to OSI and TCP/IP models. (2014).
[2] Bill Marczak and John Scott-Railton. 2020. Move Fast and Roll Your Own Crypto. Report, The Citizen Lab (2020).