API Tutorial

Manage video stream quality for small frame videos with the Daily API

Improve call performance for small frame videos



Jess Mitchell, Mattie Ruth Kramer Backman
Published: 17 Jan 2022 (Updated: 17 Jan 2022) • 5 min read



At Daily, we know call quality and performance are key factors in whether all participants involved consider the call successful.

To address this, we've previously discussed improving browser performance for video calls, as well as how your code can impact call quality. Similarly, we've reviewed how to handle growing the size of your video calls effectively.

One area where developers may be able to get another performance win is when apps are displaying small frame videos. That is to say, when the actual video being displayed in a call has small dimensions (i.e. a thumbnail) or is viewed on small screens (e.g. mobile devices). In these situations, we can often use a lower resolution for said video.

Today, we'll look at two use cases for small frame videos:

1. When everyone in the call will only ever view the video in a small frame
2. When only some of the call participants will view the video in a small frame, or video sizes will change throughout the call

Another way we can think of this is:

1. We'll only need to send a lower resolution layer of each video to other call participants, because that's the only option that will be used
2. We'll need call participants to be able to decide which layer they will receive, but not everyone will choose the same option

⚠️ ***Note: These tips are only applicable to calls in SFU mode (more than 4 participants). Relatedly, Daily Prebuilt already handles this logic for you so these tips are intended for custom Daily implementations.***

What is a simulcast layer?

When we talk about a video's "lower" layer, we're referring to one of the video track's simulcast layers.

This is described in a previous Daily blog post as follows:

> *With WebRTC simulcast, instead of sending a single track to the SFU, a publishing participant's web client sends the same source track at a few different resolutions, bitrates, and frame rates. Each of these groups of settings is known as a simulcast layer.*

By default, participants on Daily desktop video calls typically send three layers (two when in mobile) to the SFU:

`daily-js` default simulcast layers on desktop

	LAYER 0	LAYER 1	LAYER 2
Bitrate (bps)	80,000	200,000	680,000
Frame rate (fps)	10	15	30
Resolution (width x height)	320x180	640x360	1280x720

`daily-js` default simulcast layers on mobile

	LAYER 0	LAYER 1
Bitrate (bps)	80,000	520,000
Frame rate (fps)	10	30
Resolution (width x height)	320x180	1280x720

Simulcast layers only apply when connecting via an SFU and, by default, the Daily client and SFU will work to send the highest layer possible.

There are several factors that affect which layer you receive. They include:

1. **The available bandwidth on the receiving end**: The SFU will send a lower layer if the current one exceeds the available bandwidth.
2. **The send-side is not sending all the layers**: This typically occurs when the browser detects network or CPU issues or the highest video resolution of the device does not support the highest layers. It's also worth noting that the browser can and will modify the actual bitrate, framerate, and resolution sent on each layer for these same reasons, so the actual settings used may not match the configuration.

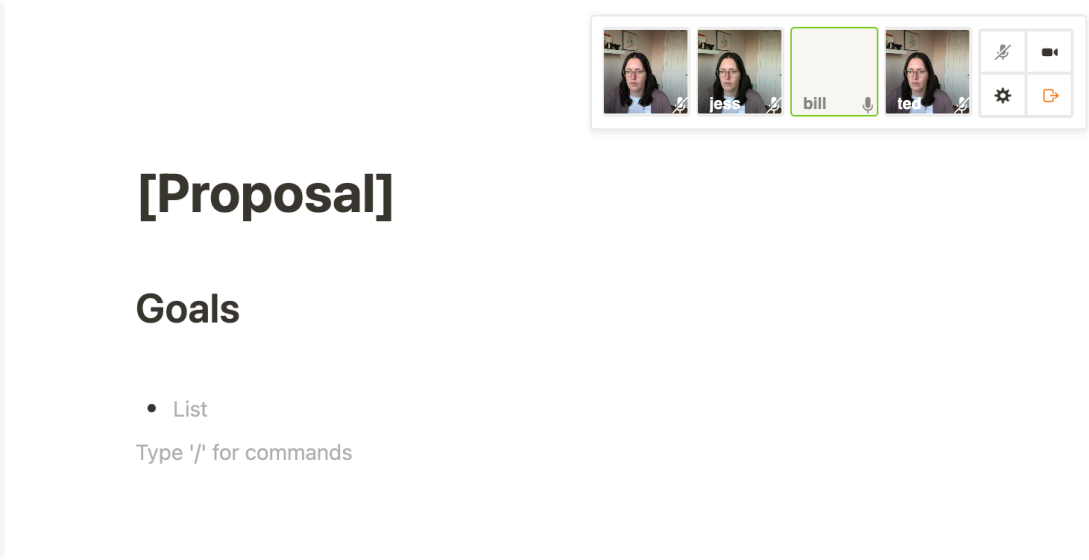
The simulcast layers that are sent can be manually updated via the Daily API when these default values are not optimal for the current use case.

⚠️ ***Note: The default values have been carefully chosen by Daily as best-overall-fit to work across all platforms and networks. As a result, changing these defaults can have negative impacts. Please contact us if you are considering changing them so we can help with this process.***

Sending only a low resolution video

In cases where we know the videos will always be small frames or low resolution, we can manually change the upstream camera video bandwidth used. This will reduce the data sent to peers via the SFU and improve overall call performance.

Another way to think about this is that it will potentially allow for more people to be in the call without a noticeable performance hit for participants.



App UI example where all videos are thumbnails and you may want to consider using `setBandwidth`

To achieve this, we can (carefully) use Daily's setBandwidth method.

Note: In case you missed the warning above, we recommend contacting

us before using `setBandwidth` specifically to ensure it is a good option for your app.

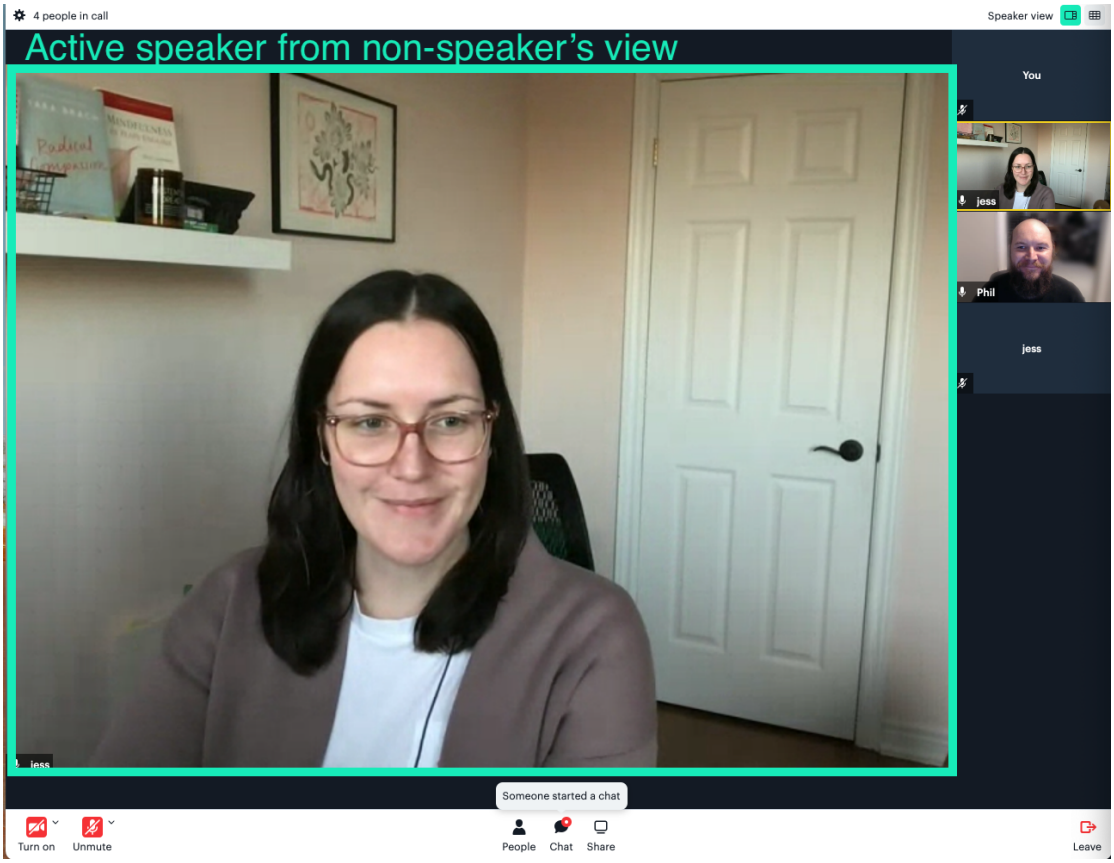
```
callFrame.setBandwidth({
  kbs: 20,
  trackConstraints: { width: 64, height: 64, frameRate: 3 },
});
```

In this example above, we're using `setBandwidth()` to transmit 64x64 images at a frame-rate of 3 images per second, with a target video bandwidth cap of 20 kilobits per second.

Receiving only the lowest simulcast layer

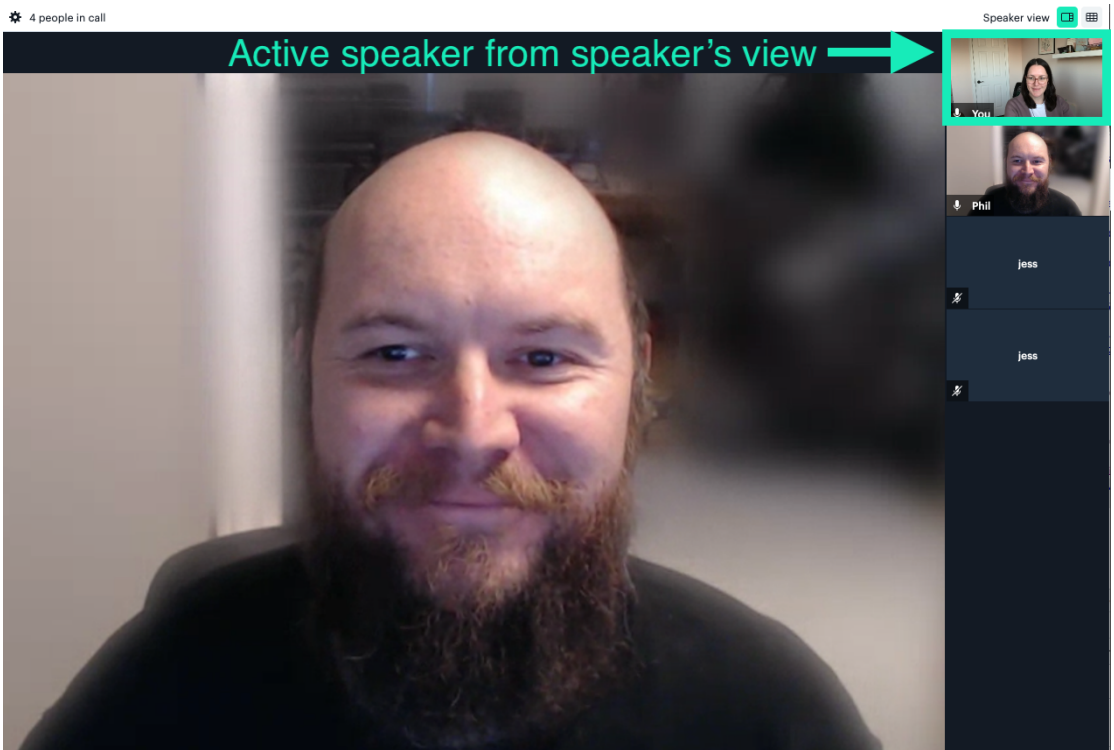
In cases where a local call participant needs to only receive the lowest resolution of a video, developers can use Daily's `updateReceiveSettings` method or the `receiveSettings` property when joining a call.

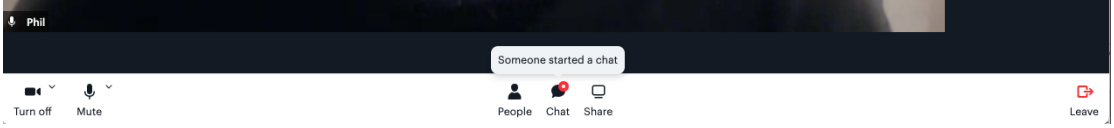
An example of when you could use this is if your app has an active speaker mode.



In Daily Prebuilt, non-active speakers view the active speaker in the largest tile

We can use the active speaker logic implemented in Daily Prebuilt's active speaker mode as an example. The active speaker will never see their own video in the large "active" video tile. This means the app will render the active speaker in the active tile for everyone in the call ***except*** the speaker, who will see a smaller version of their video in the participant bar.





The active speaker only sees their own display on the right: a small frame in the participant bar

Since the active speaker role will require a higher resolution in everyone's view except the speaker's, we want to receive the different simulcast layers and choose which to use based on app-specific logic.

```
// Set the receive settings for a participant at join time
callFrame.join({
    url,
    receiveSettings: {
        'some-participant-id': {
            video: {
                layer: 0,
            },
        },
    },
});
```

Above, we're setting a specific user's `receiveSettings` on join to receive only the lowest simulcast layer available.

To do this dynamically during a call – say, when the active speaker changes – you can use the Daily `updateReceiveSettings` method like so:

```
callFrame.updateReceiveSettings({
    'some-participant-id': {
        video: {
            layer: 0,
        },

    },
});
```

Note: You may have noticed changing the receive settings via does not have the same warning as changing the layers you send (i.e. with `setBandwidth`).

This is because — if you are choosing from the default layers Daily sets

— the layers you are choosing from are considered safe options. 👍

We hope these tips help further improve your Daily video app's performance! To learn more about how to improve Daily call performance, check out our guide on how to scale large video calls.

Get updates from Daily

Don't miss a thing! Subscribe to our monthly newsletter and we'll keep you up-to-date with the latest product releases, tutorials, and more from Daily.

Email*

Country*

Join peerConnection - our video/WebRTC developer community.

By clicking submit below, you consent to allow Daily.co to store and process the personal information submitted above to provide you the content requested.

Submit

More in API Tutorial

Implementing cursor sharing with Daily's video call API

24 Oct 2022 - 11 min read

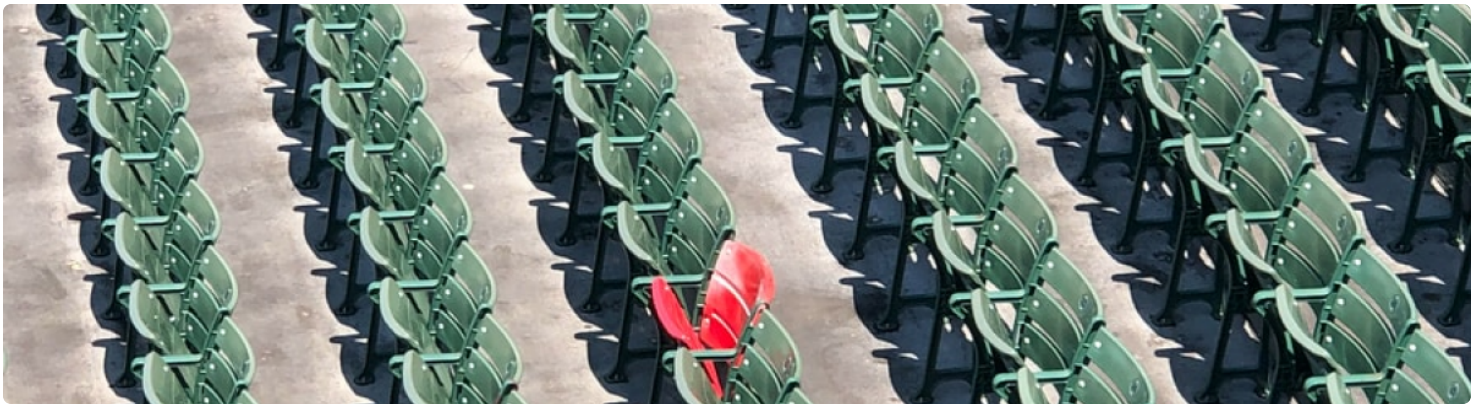
Add hand-raising to your custom video app with the Daily user data API

12 Sep 2022 - 12 min read

Stream with HLS while recording to a custom S3 bucket using Daily

30 Aug 2022 - 5 min read

See all 82 posts →



Code & Tutorials

Highlight the active speaker on a video call

Use the Daily API to showcase the participant currently talking, and minimize the rest



Kimberlee Johnson
20 Jan 2022 – 8 min read



API Tutorial

Build a call lobby to control meeting access

Use the Daily API to selectively admit participants



Kimberlee Johnson
13 Jan 2022 – 8 min read