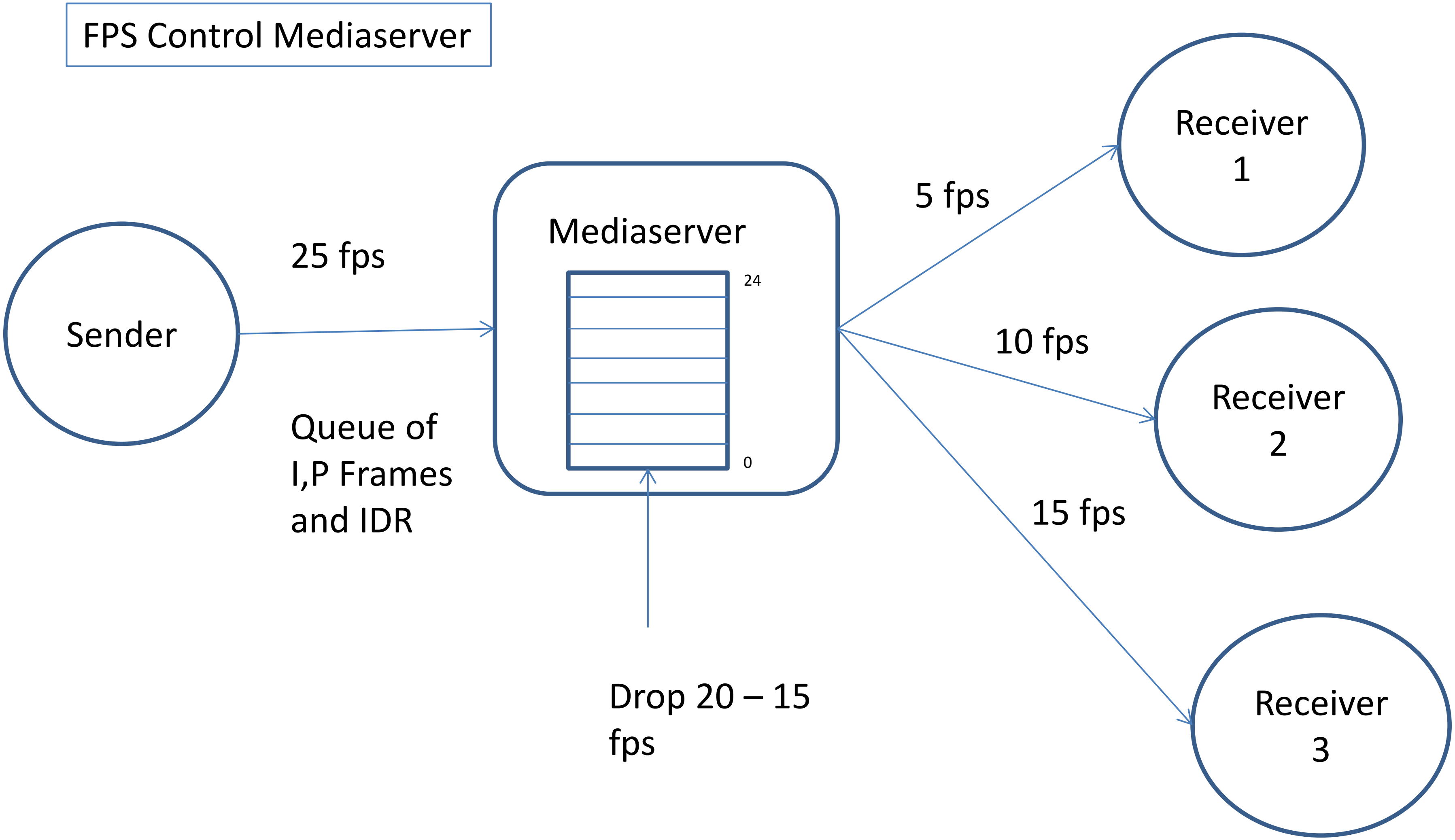

FPS Control Mediaserver
Sender
25 fps
Queue of
I,P Frames
and IDR
Mediaserver
24
0
Drop 20 – 15
fps
5 fps
10 fps
15 fps
Receiver
1
Receiver
2
Receiver
3

A. Adaption of frame rate with transport-cc statistics
B. Adaption of bit rate with transport-cc statistics, google's congestion control and bwe (Not doing it right now)
C. Adaption of screen resolution based on simulcast (Not doing it right now)

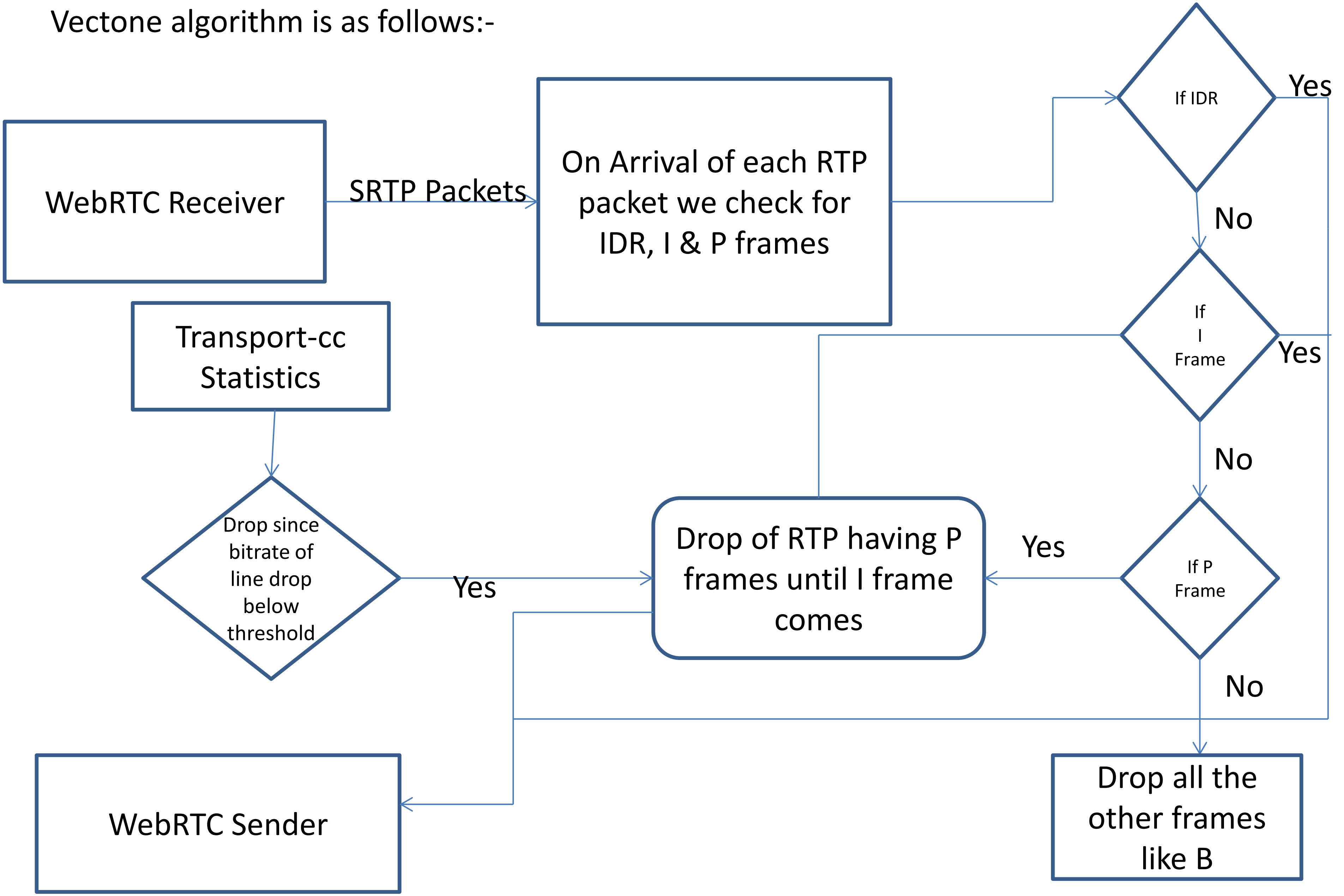
A. Adaption of frame rate with transport-cc statistics

We can do it in two ways
a) Reduce the Frame rate of sender it self. But then all receiver will get frame with very low frame rate. Which is not good

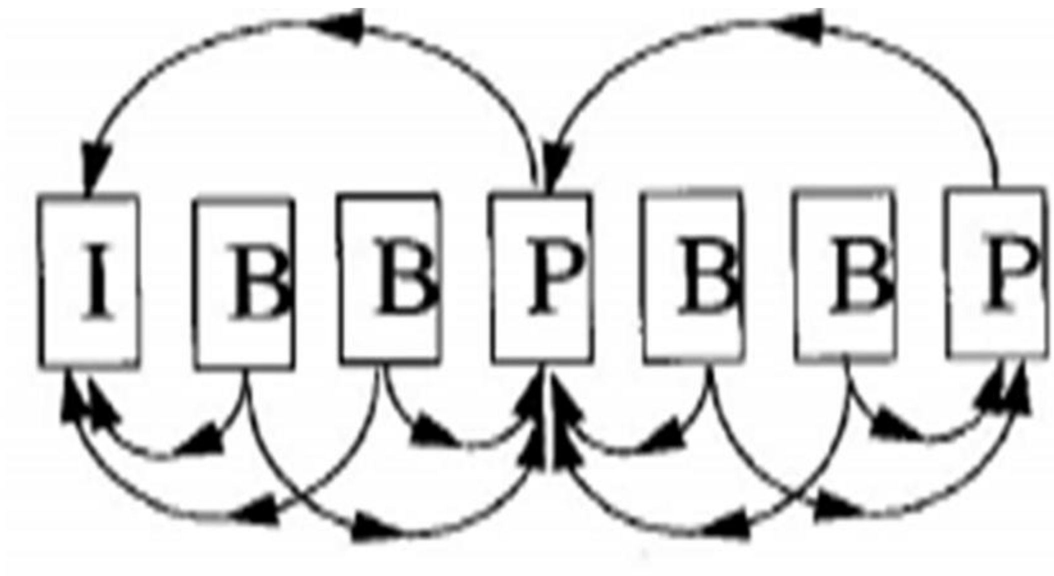
b)We will implement frame rate Adaption as shown in slide 1.

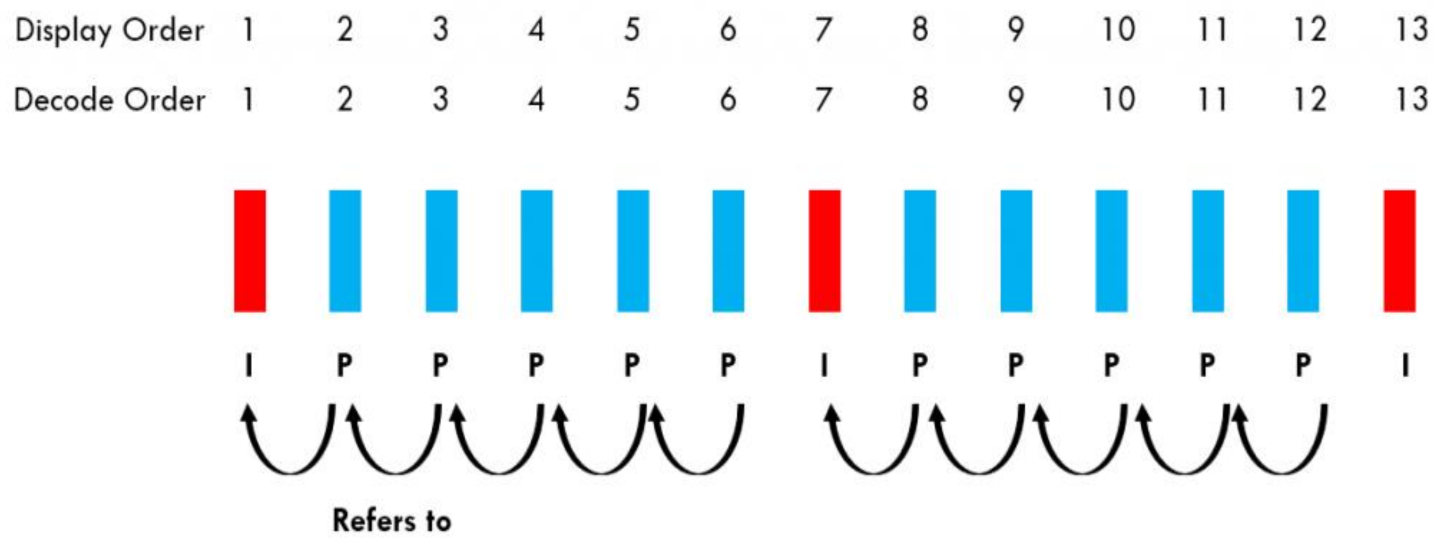
The advantage of this algorithm is if a receiver have good connectivity he will receive good frame rate even when other participants receiving lower frame rate

Vectone algorithm is as follows:-



B frames will never comes in video calls, since we use strict base line profile H264





This is the only possible situation in video call

The P-frames refer to previously encoded I/P-frames as discussed earlier.
You can also see that the order in which the frames are encoded/decoded is the same as how they are presented to the user. This is because P-frames only refer to previously encoded pictures.

WebRTC receiver receives the RTP packets from sender

On arrival of each RTP packets we peek inside RTP to check for IDR, I and P frames without decoding frames

Only trailing P frames will be drop if the bitrate of peer connection drop below threshold

To peek inside Secured RTP we need certificates.