

Use case	offerToReceive + createNullMedia	createRtpSender + createRtpReceiver	addMedia + RtpTransceiver		
answerer rejects offered tracks	<code>receiver.track.stop();</code> <code>pc.createAnswer();</code> There's no way to know!	<code>receiver.stop();</code> <code>pc.createAnswer();</code> Know via <code>sender.stopped</code>	<code>media.stop();</code> <code>createAnswer();</code> Know via <code>media.stopped</code>		
warmup	<code>pc.addTrack(</code> <code>pc.createNullMedia("video"))</code> <code>pc.createOffer();</code>	<code>pc.createRtpSender("video");</code> <code>pc.createRtpReceiver("video");</code> <code>pc.createOffer();</code>	<code>pc.addMedia("video");</code> <code>pc.createOffer();</code>		
answerer sends more than offerer	<code>pc.addTrack(track);</code> <code>pc.createOffer(</code> <code>{offerToReceiveVideo: 2});</code>	<code>pc.addTrack(track);</code> <code>pc.createRtpReceiver("video");</code> <code>pc.createOffer();</code>	<code>pc.addTrack(track);</code> <code>pc.addMedia("video",</code> <code>{send: false});</code>		
offerer disallows receiving	<code>pc.addTrack(track);</code> <code>pc.createOffer(</code> <code>{offerToReceiveVideo: 0});</code>	<code>pc.createRtpSender(track);</code> <code>pc.createOffer();</code>	<code>pc.addMedia(track,</code> <code>{receive: false});</code> <code>pc.createOffer();</code>		
stop sending, the other sides knows	<code>pc.removeTrack(sender.track);</code> <code>pc.createOffer();</code> <b>There's no way to know!</b>	<code>sender.stop();</code> <code>createOffer();</code> Know via <code>receiver.stopped</code>	<code>media.stop();</code> <code>createOffer();</code> Know via <code>media.stopped</code>		
stop receiving, the other sides knows	<code>receiver.track.stop();</code> <b>There's no way to know!</b>	<code>receiver.stop();</code> Know via <code>sender.stopped</code>	<code>media.stop();</code> Know via <code>media.stopped</code>		
know when remote endpoint kills m-lin	<b>There's no way to know</b>	Know via <code>sender.stopped</code> and <code>receiver.stopped</code>	Know via <code>media.stopped</code>		
Issue 187	???	Stopped senders and receivers never are put back into the offer.	Stopped media never goes back in the offer.		
matching senders to receivers	Implicit (browser manages)	<code>pc.createRtpSender("video");</code> (n times) <code>pc.createOffer();</code> <code>pc.createRtpReceiver("video");</code> what's the rule?	Explicit		
forcing unidirectional m-lines	Possible for certain cases only	Not clear (explicit MID settin	<code>pc.addMedia(send:false</code> <code>pc.addMedia(send:true,</code>		
receive media in race with answer	Not possible	<code>pc.createRtpReceiver("video");</code>	<code>pc.addMedia(send:false, receive:true);</code>		
Pros	Minimal changes to spec	No ugly hacks or kludges Know when sender/receiver stopped	No ugly hacks or kludges Know when sender/receiver st		
Cons	<code>createNullMedia</code> is kind of hacky <code>offerToReceiveVideo</code> is kind of kludgy inconsistent: <code>track.stop()</code> vs. <code>removeTrack</code> don't know when sender/receiver stoppe	Adds three methods and one attribute Deprecates <code>offerToReceiveX</code>	Adds two methods and one ob Deprecates <code>offerToReceiveX</code>		