#### 四、基于Mesh架构的例子

目前我们实现了一个基于Mesh架构的多人视频通话Demo应用，基于这个应用测试了在不同情况下多人视频通话的情况。  
WebRTC不同Sent分辨率下的最大码率：  
webrtc\media\engine\webrtcvideoengine2.cc  
// The selected thresholds for QVGA and VGA corresponded to a QP around 10.  
// The change in QP declined above the selected bitrates.  
static int GetMaxDefaultVideoBitrateKbps(int width, int height) {  
if (width \* height <= 320 \* 240) {  
return 600;  
} else if (width \* height <= 640 \* 480) {  
return 1700;  
} else if (width \* height <= 960 \* 540) {  
return 2000;  
} else {  
return 2500;  
}  
}  
最大默认码率的意义在于：在特定分辨率下，只要视频流的发送码率达到最大默认码率，视频质量已经足够好，再增大发送码率对视频质量的影响已经不大。

###### 1、H264软件编解码

（目的是观察CPU对视频流Sent分辨率的影响）  
参与方  
A - We45  
B - 小米Note  
C - 红米

发送端Input分辨率都设为1280x720  
1.在只有A和B的情况下，各Sent分辨率如下：  
A->B 640x360  
B->A 640x360  
Sent分辨率降低都是由CPU导致的。  
2.C加入后，各Sent分辨率如下：（Sent分辨率实际上是会变化的，下面的数据是一段时间之后基本稳定的数据，视频通话继续进行Sent分辨率还有可能进一步降低）  
A->B 480x270  
A->C 480x270  
B->A 320x180  
B->C 320x180  
C->A 320x180  
C->B 320x180  
Sent分辨率降低都是由CPU导致的。

由此可以得出如下结论：使用软件编解码，影响Sent分辨率的主要因素是CPU负载；参与多人视频通话的参与方越多，WebRTC Sent分辨率越低。

###### 1、H264硬件编解码

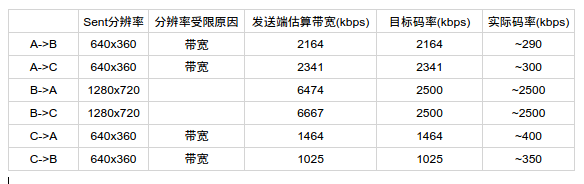
（目的是排除CPU对视频流的影响，方便观察带宽对视频流的影响）  
参与方  
A - We45  
B - 小米Note  
C - We45

1.在只有A和B的情况下，各项数据如下：



A和B

2.C加入后，各Sent分辨率和导致分辨率降低的原因如下：



A、B和C

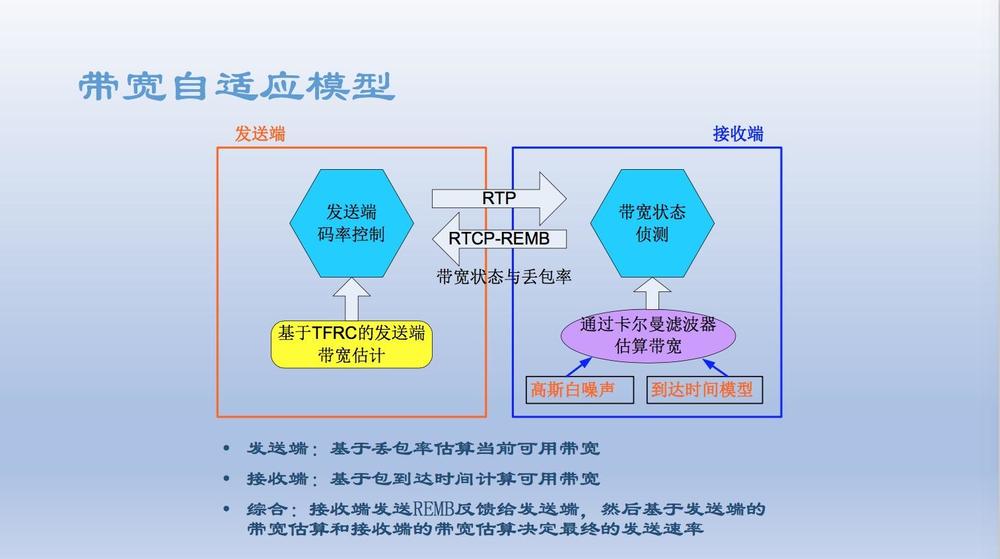
由此可以发现如下现象：  
1.目标码率由发送端估算带宽和分辨率对应的默认最大码率决定，但实际码率不一定与目标码率一致。如果实际码率低于发送端估算带宽，则Sent分辨率会降低。如果实际码率等于发送端估算带宽，Sent分辨率不会降低。

疑问1：实际码率是由什么决定的？

实际码率除了收到发送端目标码率的影响，还有可能收到接收端接收码率的影响。按照目标码率发送数据，但因为接收端的接收能力的限制，接收端并不足以即时接收数据，发送端收到接收端的反馈信息后，会做出相应的调整，降低实际的发送码率。这就是WebRTC的带宽自适应算法：

WebRTC中的带宽自适应算法分为两种：  
１.发端码率控制，　原理是由rtcp中的丢包统计来动态的增加或减少码率，在减少码率时使用TFRC算法来增加平滑度。  
２.收端带宽估算，　原理是收到rtp数据，估出带宽；用卡尔曼滤波，对每一帧的发送时间和接收时间进行分析，从而得出网络带宽利用情况，修正估出的带宽。

两种算法相辅相成，收端将估算的带宽发送给发端，发端结合收到的带宽以及丢包率，调整发送的码率。



WebRTC带宽自适应算法