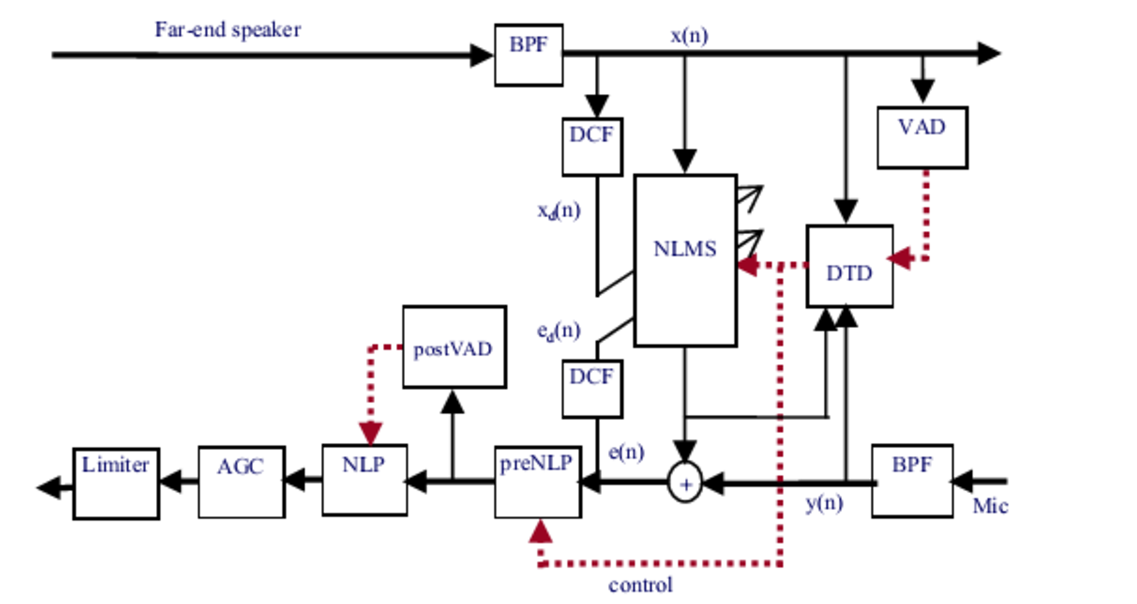
WebRtc之AEC回声消除算法

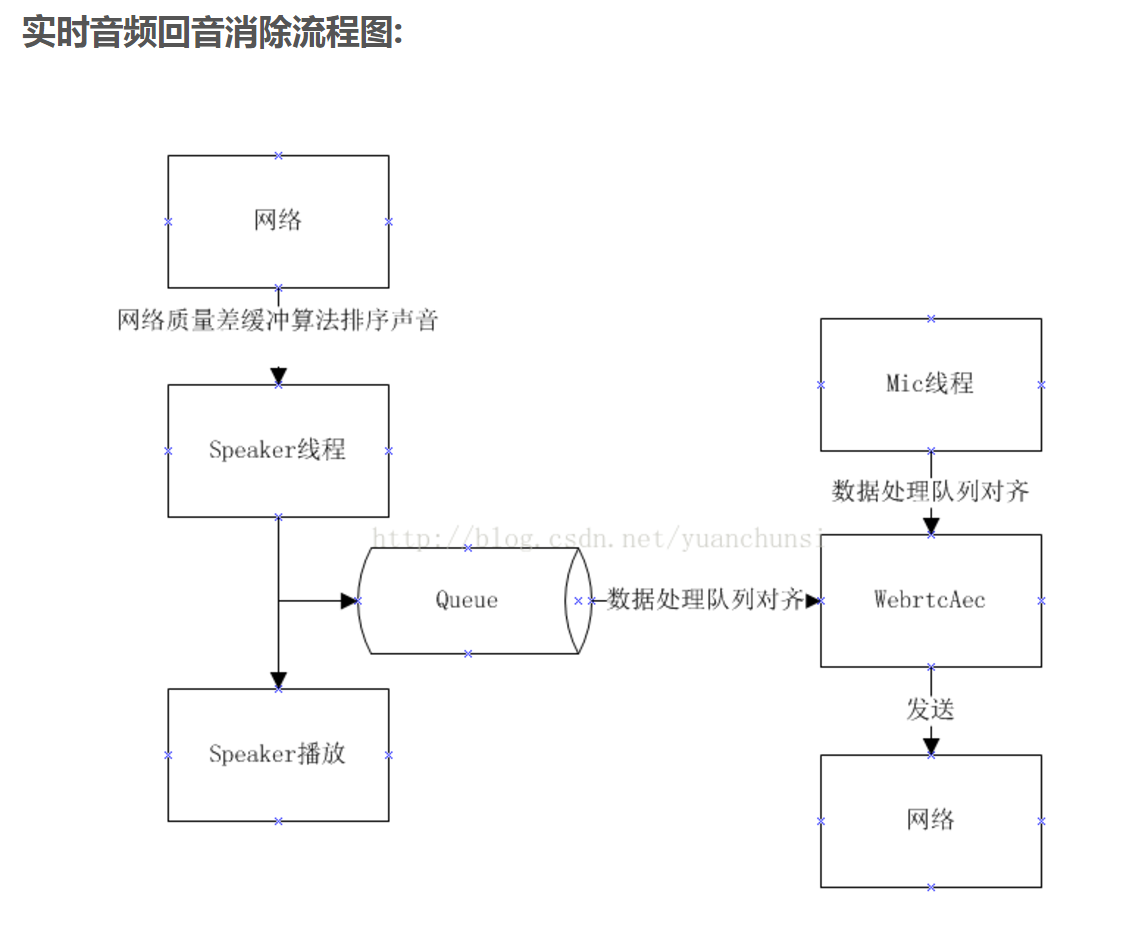
1. 概述

WebRtc中的AEC算法属于分段块频域自适应滤波算法，Partioned block frequeney domain adaptive filter(PBFDAF)。



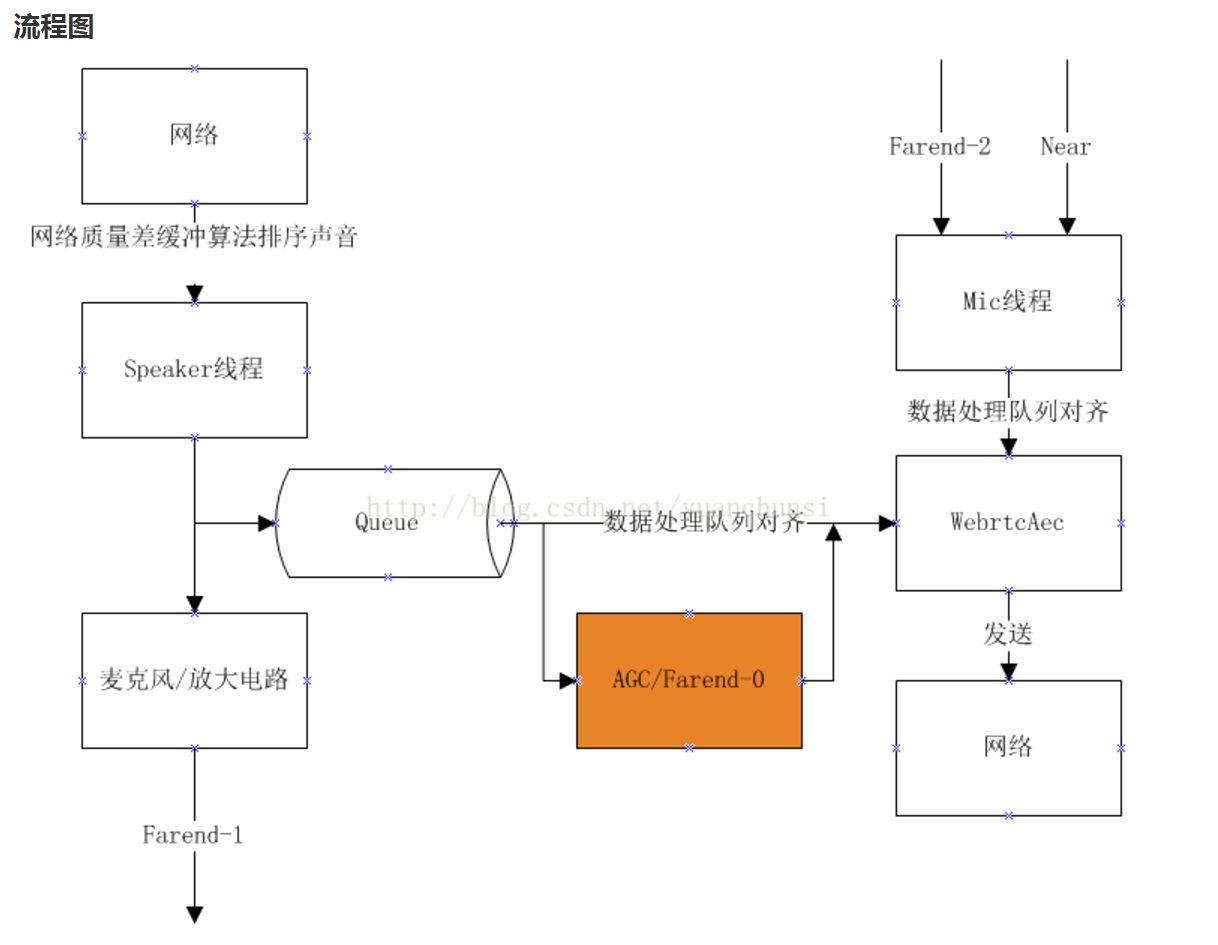
上图是功能框图，BPF是Band Pass Filter，用来滤掉远端信号中的过高和过低的频率分量（类似降噪？），DCF是correlation filter，用来使得NLMS快速收敛的。VAD是监测远端是不是有声音信号的，NLP是用来去掉残余回声的。

没有一个好的DTD。这就造成没有DT的时候消除回声很干净，有DT的时候，近端talk也被消掉了。

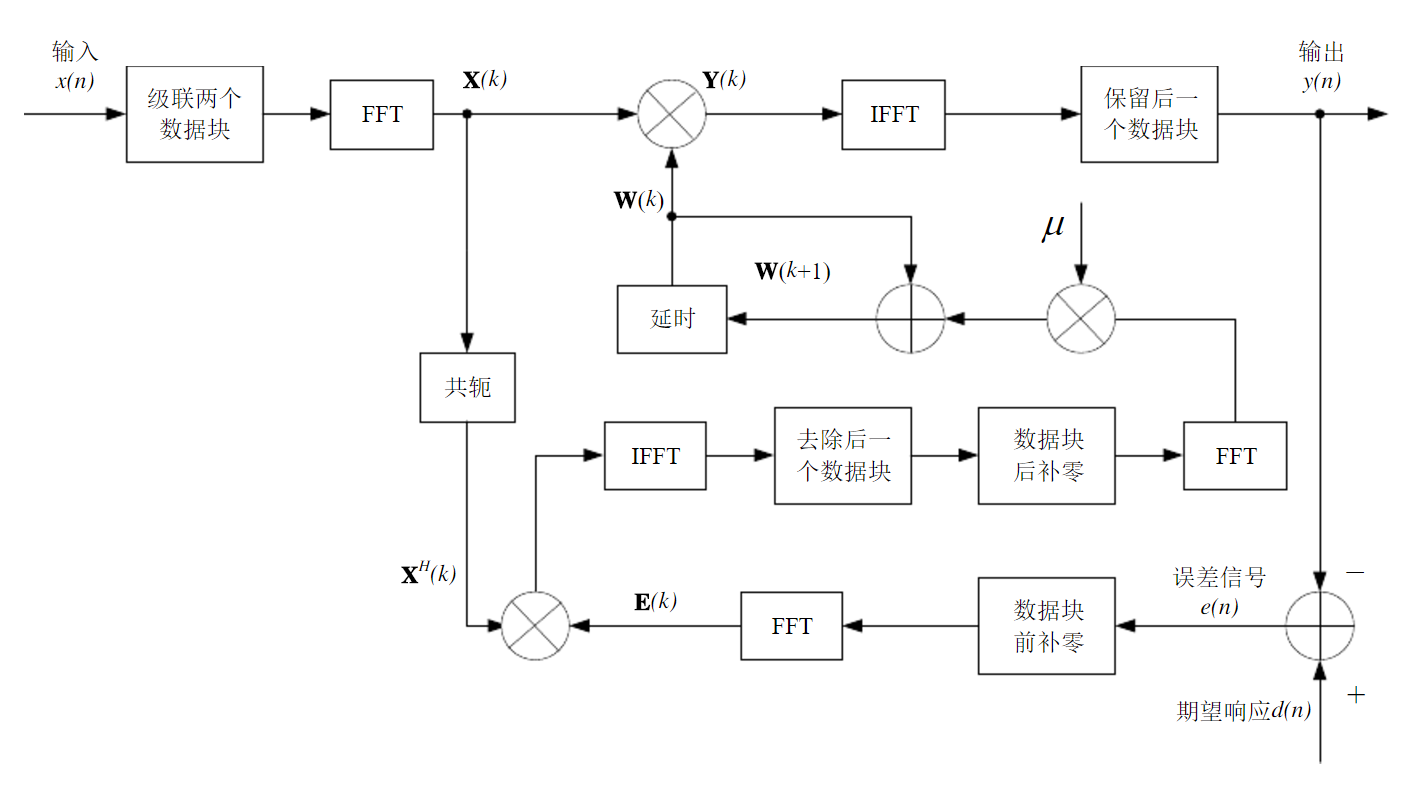


**关键知识点：**

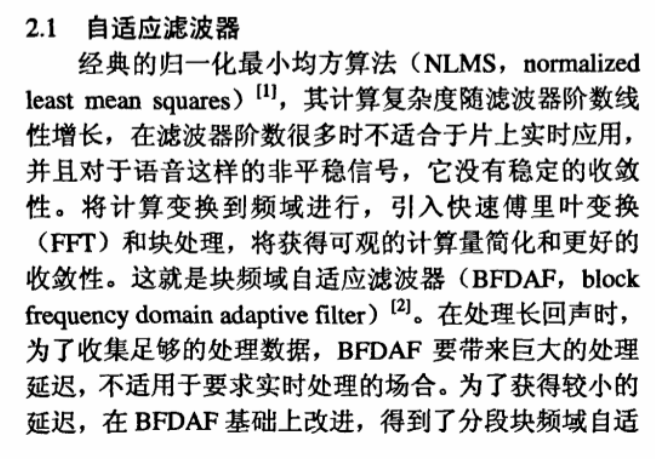
1. 回音消除目的是实现双向对讲(double talk)。
2. 硬件要求，低端设备视频音频加AEC后CPU（99%），声音一卡一卡，AEC没法运算。目前只做双向语音对讲，关闭设备视频录像，解决。

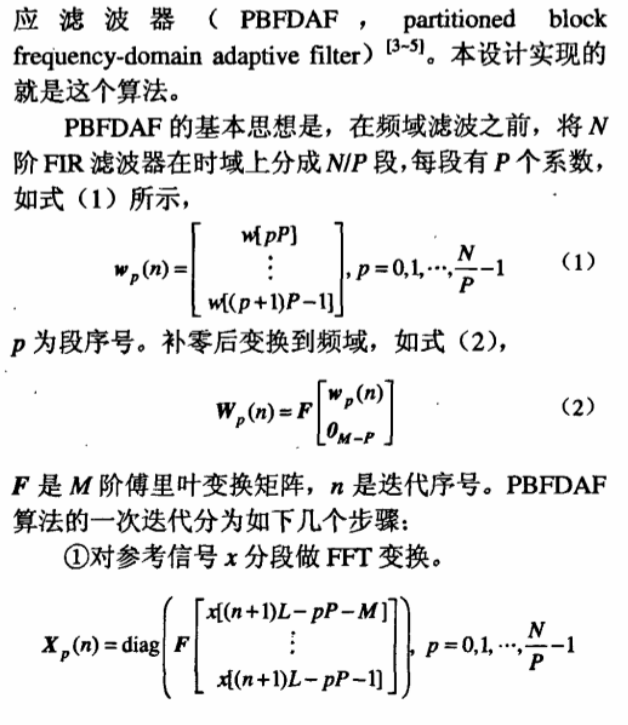
****

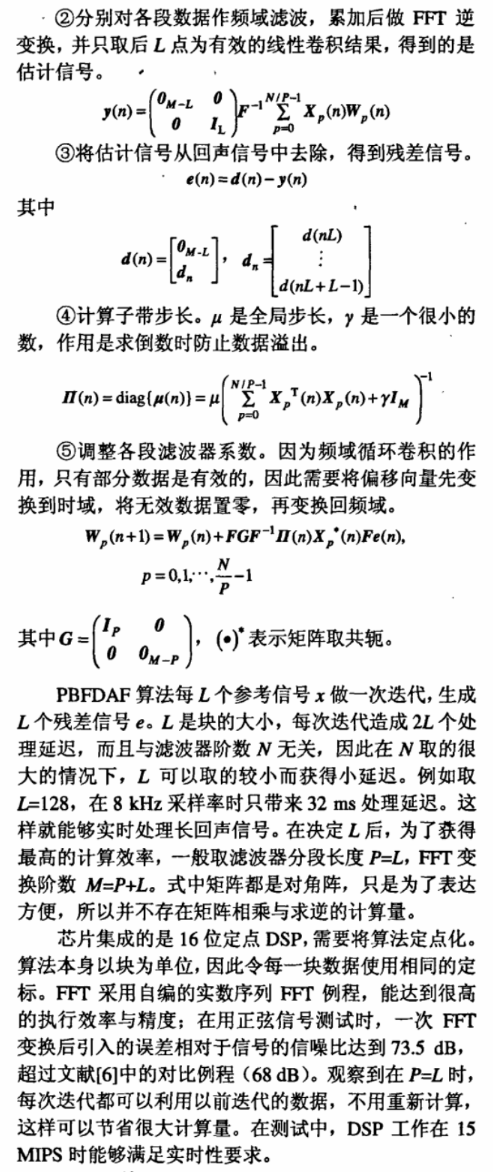
**备注：通过AGC增大Farend-0的增益，使得Farend-0与Farend-2+Near的能量相当，其中Farend-1大于Farend-2。**

****

**频域块LMS自适应滤波器结构**

****

****

****

由于工作需要，最近一直在研究WebRTC里的AEC算法。根据源码里面的fullaec.m文件，

总体来说，我认为该AEC算法是属于分段快频域自适应滤波算法，Partioned block frequeney domain adaPtive filter(PBFDAF)。具体可以参考Paez Borrallo J M and Otero M G

使用该AEC算法要注意两点：

1）延时要小，因为算法默认滤波器长度是分为12块，每块64点，按照8000采样率，也就是12\*8ms=96ms的数据，而且超过这个长度是处理不了的。

2）延时抖动要小，因为算法是默认10块也计算一次参考数据的位置（即滤波器能量最大的那一块），所以如果抖动很大的话找参考数据时不准确的，这样回声就消除不掉了。

1. 流程

(一)ProcessStream()

ProcessStreamLocked()

Agc\_manager->AnalyzePreProcess();

Beamformer->AnalyzeChunk();

High\_pass\_filter->ProcessCaptureAudio();

Gain\_control->AnalyzeCaptureAudio();

Echo\_cancellation->ProcessCaptureAudio();

Noise\_suppression->ProcessCaptureAudio();

Voice\_detection->ProcessCaptureAudio()

（二）Echo\_cancellation->ProcessCaptureAudio()

WebRtcAec\_Process()

ProcessExtended()或者ProcessNormal()

WebRtcAec\_GetSkew()

AEC enable之前，要使系统稳定，系统稳定后，填充far-end buffer.

EstBufDelayNormal()

WebRtcAec\_ProcessFrames()---包括6个步骤，第4步ProcessBlock().

WebRtcAec\_system\_delay()

(三) ProcessBlock()

1. WebRtc\_AddFarSpectrumFloat()---Block wise delay estimation used for logging.
2. EchoSubtraction()---Perform echo subtraction.

WebRtcAec\_FilterFar()---Produce echo estimate in frequency domain..

1. NLP：

EchoSuppression()

PartitionDelay()---Measures the energy in each filter partion and returns the partion with highest energy.

WebRtcAec\_UpdateCoherenceSpectra()---Updates smoothed Power Spectral Densities(PSD) and the filter diverge state is determined.

WebRtc\_ComputeCoherence()

FormSuppressionGain()---track the local filter minimum to determine suppression overdrive.

WebRtcAec\_Suppress()

ComfortNoise()---Add comfort noise.

If(exists H band)

GetHighbandGain()---H band gain average nlp over low band.

1. 细节（函数处理的输入和输出及备注）

Echo\_cancellation->ProcessCaptureAudio(AudioBuffer\*, stream\_delay\_ms());

WebRtcAec\_Process(void\* aecInst, const float\* const\* nearend, size\_t num\_bands, float\* const\* out, size\_t nrOfSamples, int16\_t msInSndCardBuf, int32\_t skew)

备注：msInSndCardBuf是输入输出的时延，就是远端信号从被reference到被aec处理之间的时间差。<http://www.2cto.com/kf/201609/546350.html>

ProcessNormal(Aec\* self, const float\* const\* nearend, size\_t num\_bands, float\* const\* out, size\_t nrOfSamples, int16\_t msInSndCardBuf, int32\_t skew)

ProcessExtended(Aec\* self, const float\* const\* near, size\_t num\_bands, float\* const\* out, size\_t num\_samples, int16\_t reported\_delay\_ms, int32\_t skew)

WebRtcAec\_ProcessFrames(AecCore\* aec, const float\* const\* nearend, size\_t num\_bands, size\_t num\_samples, int knownDelay, float\* const\* out)