

# WEBRTC Audio-related Neteq (iii): Access packet and delay calculation

Last Update:2018-09-18

Source: Internet

Author: User

Tags flush (/zqpop/flush\_36982.html)

Developer on Alibaba Coud: Build your first app with APIs, SDKs, and tutorials on the Alibaba Cloud. Read more > (<https://developer.alibabacloud.com>)

In the previous article (WEBRTC Audio-related Neteq (ii): Data structure) Neteq the main data structures, to understand the mechanism of Neteq lay a good foundation. This article is mainly about how the RTP packets received from the network in the MCU are put into packet buffer and taken out from packet buffer, as well as the calculation of the network delay value (optbuflevel) and the jitter buffer delay value (bufflevelfilt). Let's see how RTP voice packets are put into packet buffer.

said earlier that there was a slot concept for putting the RTP packets received from the network into packet buffer. Each slot has a package attribute (such as timestamp, sequence number, and so on) and payload. Packet buffer initializes the Numpacketsinbuffer (number of packages) and Insertposition (where the next package is placed) to 0, The properties and payload are also set to a reasonable value (for example, payloadlengthbytes (payload length) is set to 0, payloadtype (load type) is placed 1). When a RTP packet is to be put into packet buffer, the first step is to see if the Packetbuffer is empty (that is, if the Numpacketsinbuffer is 0). If empty, place the package directly in slot 0 (the properties of the package and the payload are placed in slot 0). If not empty, insertposition plus 1, see if there is a packet on this slot (flag is payloadlengthbytes is zero, zero means no package.) When the packet on this slot is taken, Payloadlengthbytes will be set to 0). If the package indicates that packet buffer is full, reset (called Flush in the Code) packet buffer, and then place the current package in slot 0 (the number of packets that can be placed in the packet buffer is a large value, usually not full). If it is not full, it is placed directly on the next slot.

Next see how to take a voice packet from Pakcet buffer, which depends on the timestamp value (recorded as TIMESTAMP\_FROM\_DSP) from the DSP module. Traverse packet buffer in each slot, if the timestamp of the voice packet on this slot is less than TIMESTAMP\_FROM\_DSP, and there is payload on the slot, it can be thought that this package should be actively discard out too late, Includes reset this slot and the number of packets in the packet buffer minus one. After traversing packet buffer, put the timestamp of the voice packet closest to TIMESTAMP\_FROM\_DSP (that is, the timestamp of the voice packet minus Timestamp\_from\_ The value of the DSP is the smallest) corresponding slot as the slot to be taken out, the voice packet from the slot after the same reset this slot and packet buffer packet number minus one.

As can be seen from the above description of the way the package is relatively simple, according to the voice packet to the NETEQ sequence in the buffer, and may be chaotic (when the packet is received may be disorderly). This requires that when the packet is fetched to traverse buffer to remove the timestamp from the DSP module to the nearest timestamp packet, traverse to use for loop, which increases the amount of computation. This is quite different from the design of the jitter buffer I used to do. The idea is to put a disorderly order after the package is placed in the buffer, the

packet will not need to traverse the buffer, but from the beginning to take back. Specifically how to achieve can look at my previous article (audio transmission of the jitter buffer design and implementation).

Here's how to calculate the network delay statistics (OPTBUFLEVEL), which is one of the difficulties. Assuming 20Ms per pack, ideally every 20Ms receives a voice packet from the Web. The reality is that the network has delayed packet jitter, resulting in not receiving a packet every 20Ms, but sometimes dozens of or even 100 milliseconds to receive a package, sometimes 20Ms received several packets. We want to calculate the statistical value of the network delay, as one of the basis for generating control commands to the DSP. How do you calculate it? NETEQ is calculated using the packet-to interval, which means that the current received packet is relative to the last packet received, in the number of packages. When each packet is received, the Packetiatcountsamp (number of sampled units) is cleared, and each frame of the data played at a later time to the Packetiatcountsamp plus a frame of sample points (in AMR-WB per frame 20Ms For example, each frame has 320 sampling points. Each frame, Packetiatcountsamp increases by 320), the next time a package to, take packetiatcountsamp divided by 320 can calculate the interval between two packets.

The algorithm for calculating the network delay is given below:

1, calculate the current packet absolute arrival interval IAT (in terms of data packet number units), the formula is as follows:

According to the formula, the early arrival of the IAT of the packet is 0, the normal arrival of the IAT is 1, delay the arrival of a packet time the IAT is 2. The IAT has a maximum value of 64, which means there are 65 (0-64) possible types.

2, update the probability distribution of the IAT on each value (0-64). The probability of each value (0-64) at initialization is 0, and as the packet arrives, the probabilities on each value change dynamically. The probability update is divided into the following small steps:

1) forgetting the current probability with the forgetting factor F, the formula is as follows:

Here is the concept of a forgetting factor (forgetting factor). The probabilities on each value are calculated and the new probabilities are obtained.

2) Increase the probability of this calculation to the IAT, the calculation formula is as follows:

3) Update The forgetting factor F, so that f is an increasing trend, that is, the longer the talk time, the more stable the probability distribution of the IAT in the packet interval. The calculation formula is as follows:

4) Adjust the probability of this calculation to the IAT so that the probability distribution of the whole IAT is approximately 1. Assuming that the sum of the current probability distributions is tempsum, the formula is as follows:

3, the IAT value that satisfies 95% probability is counted as B. The value of B can be calculated according to the following formula.

4, statistics the peak of IAT

The NETEQ uses two arrays of length 8 to count the peak value of the IAT, one to store the peak amplitude and the other to store the peak interval. The peak interval is another parameter Peakiatcountsamp in the structure AutoMode, which is used to count the interval of the peak distance that is currently detected, measured in number of samples. When the value of the IAT is greater than 2B, it is assumed that the peak appears, and the current IAT and Peakiatcountsamp values are present in the array. If the array is not full, it is placed in an empty position after the last peak position; If full, the oldest peak in the array is eliminated, the other peaks shift to the left, and the new peaks are placed in the array where index is 7. It should be explained here that when the value of two arrays is less than 8, the peak array does not work.

5, calculate optbuflevel

When the peak array is in effect and the current peakiatcountsamp is less than or equal to twice times the maximum interval in the peak interval array, optbuflevel takes the maximum value in the peak array. Otherwise the optbuflevel will be B.

The above is my scripted how to calculate the network delay algorithm expressed. I basically understand the idea of the algorithm, but it is unclear how some of the coefficients in the algorithm are obtained. UseGoogleSearch, did not find the relevant documentation, which is also a lot of open source software common problem, no documentation. My guess is the coefficient value that the relevant developers get by using mathematical modeling methods. If any friend knows, trouble to tell, thank you first. Tell me about my understanding of this algorithm. The network delay is calculated based on probability, and there are 65A sample (0Delay, a packet delay, 2A packet delay, ....., 64Delay). The probability of each sample at initialization (%) is 0. After the call some delay value appears its probability value will become larger, the corresponding other delay value of the probability will be smaller (already zero no way to smaller, still 0). In the algorithm, the forgetting factor is used to reduce the probability of each delay value, then the probability of the time delay value is changed, in order to guarantee the probability and 1 To do some fine tuning (also to update the forgetting factor). Then the probability of each delay value is added from the zero delay to achieve The value of the 95% can be considered as the delay value initially. For example, 0 delay probability is 0.1, 1 packet delay probability is 0.7, 2 packet delay probability is 0.09, 3 packet delay probability is 0.07, then the probability and 0.96, has reached the 0.95 line, take the network delay maximum value of 3, you can initially think that the network delay of 3 packets delay. Also depends on the current network conditions, if a period of time delay peak, indicating that the current network environment is worse, in order to improve the voice quality needs to increase the value of network delay, the peak array of the maximum value, as the ultimate network delay value.

The network delay is calculated after the voice packet is put into packet buffer. The jitter buffering delay is received after the DSP module sends the feedback information to the MCU module (to use the feedback information) and before the voice packet is taken from packet buffer. The calculation steps are given below:

1, according to the number of voice packets already in packet buffer to calculate the number of samples, recorded as Samples\_in\_packetbuffer, which depends on the sampling rate and packet length, taking AMR-WB as an example, the sampling rate is 16kHz, the packet length is 20ms, It can be calculated that there are 320 samples per pack. Assuming that there are 5 voice packets in packet buffer, the number of samples already in packet buffer is 1600 ( $1600 = 320 \times 5$ ).

2, samples that are not played in speech buffer (i.e., sampleleft) are also counted in the jitter buffer delay. its addition to the number of samples in packet buffer is the real-time jitter buffer delay (samples\_jitter\_delay, in number of samples), i.e.  $\text{Samples\_jitter\_delay} = \text{Samples\_in\_packet\_buffer} +$

samplesleft, divided by the number of samples per packet Samples\_per\_packet, you can get real-time jitter buffer delay value (in the number of packets).

3, calculate the bufferlevelfilt, the formula is as follows:

Here is calculated jitter Buffer delay Adaptive average, F is to calculate the mean of the forgetting factor, according to the network conditions of adaptive changes, the specific value is shown below:

where b is the B value in front of the network delay (in number of packets).

4, if the acceleration or deceleration play, you need to modify the Bufferlevelfilt, the formula is as follows:

Where samplesmemory represents the acceleration or deceleration of the data length after the scaling changes, the number of samples is units. If the acceleration, the samplesmemory is positive, bufferlevelfilt decrease, if the deceleration, samplesmemory negative, bufferlevelfilt become larger.

The above describes the MCU network delay and jitter buffer delay calculation, MCU also received the DSP module sent feedback report. The next MCU will be based on these to determine what kind of control command (acceleration/deceleration, etc.) for the DSP module, which is the main content of the following article.

This article is an English version of an article which is originally in the Chinese language on aliyun.com and is provided for information purposes only. This website makes no representation or warranty of any kind, either expressed or implied, as to the accuracy, completeness ownership or reliability of the article or any translations thereof. If you have any concerns or complaints relating to the article, please send an email, providing a detailed description of the concern or complaint, to info-contact@alibabacloud.com. A staff member will contact you within 5 working days. Once verified, infringing content will be removed immediately.