CHAPTER 4

Improvement Mechanism

In this chapter, we describe how to operate the CAC mechanism after obtaining the MD from Markov Chain model to constrain the number of connections. Then we use a Token Bucket mechanism to ensure if there is a burstness which will not affect our QoS guarantee. Finally, we employ Packet Drop mechanism to improve throughput.

4.1. Call Admission Control (CAC)

If we want to reach QoS guarantee, we need to use CAC mechanism to constrain the connections. When there are more and more nodes want to transmit with SSQAP, the delay time will increase rapidly. Finally, it will be over delay requirement consequential. Thus, we need to constrain the number of QSTA connections.

4.1.1. Delay requirement

After we get the delay from Markov Chain, we can make two tables. For example, we get both VO&VI adds a connection delay as Table 4-2 (Table 4-1 is for comparison). And we have known the relationship with Math and Qualnet (from Figure 3-7. Just VI traffic: Qualnet divide Math is about 1.4. Both VO and VI traffic: Qualnet divide Math is about 2). Now, there are 6 VI traffics connecting with SSQAP, which senses the delay is 8.7. If there is a new VI connection request arrival, SSQAP calculate as follow:

If SSQAP accept a new VI connection: 8.7 + (7.3 – 6.3)*1.4 = 10.1

In Qualnet: the actually delay is 10.0

Thus, the new connection will be accepted. For another example which is a VO connection. SSQAP can calculate as follow:

If SSQAP accept a new VO connection: 2*(5.7 – 4.8) – 1.4*(7.3 – 6.3)

= 1.8 – 1.4 = 0.4

In Qualnet: (11.5 – 9.8) – (10.0 – 8.7) = 1.7 – 1.3 = 0.4

Thus, we assume if the new connection is accepted, the delay has to add 0.4 ms.

Table 4-1: The delay from Qualnet.

Number	VO & VI	VI
6	9.8	8.7
7	11.5	10.0

Table 4-2: The delay from Markov Chain.

Number	VO & VI	VI
6	4.8	6.3
7	5.7	7.3

4.1.2. Bandwidth requirement

After SSQAP estimate the delay requirement and if the delay is satisfied, SSQAP need to send bandwidth requirement to BS and wait for granting. SSQAP will accept the new connection when the BS grants the bandwidth requirement.

The CAC algorithm as follows:

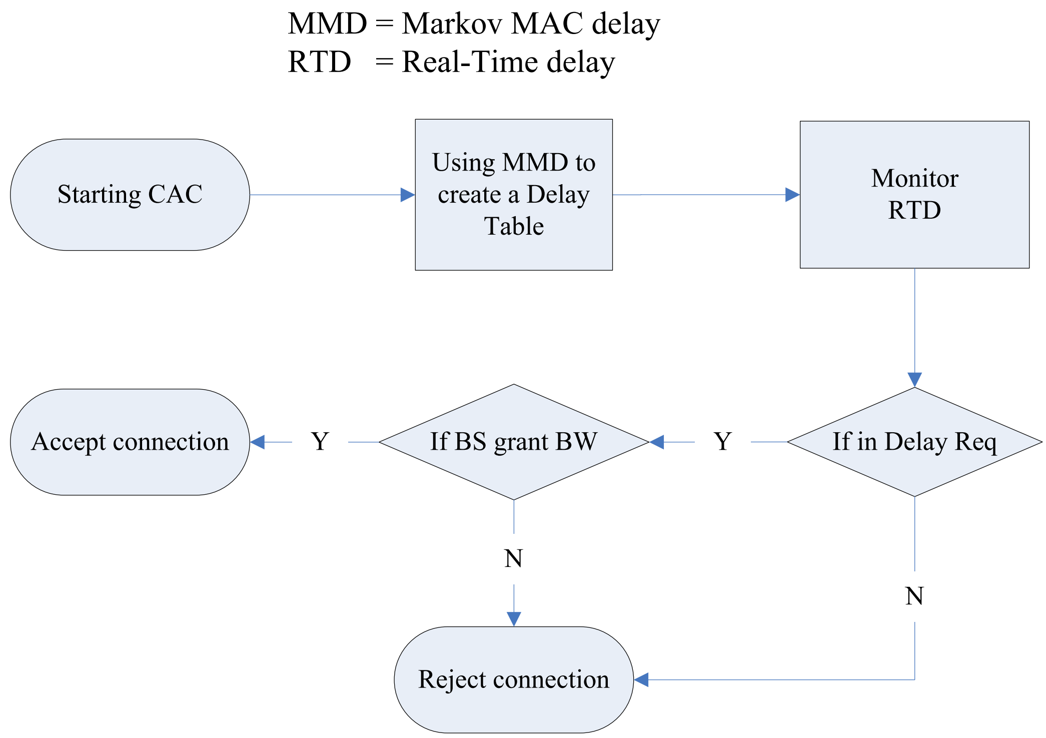

Figure 4-1: The CAC algorithm

4.2. Token Bucket Mechanism

We can estimate the connection number and constrain the traffics arrival to SSQAP. Now, we use Token Bucket mechanism to control the VI (nrtPS) output rate for avoiding burstness. But we do not just use it. The process about Token Bucket is as follows: We will describe our Token Bucket parameters first. The next step is to initialize the token rate r and bucket size B. When every packets arrival SSQAP, it will monitor and obtain some information to use it for tuning the token rate r and bucket size B.

4.2.1. Token Bucket Parameters

We briefly introduce the parameters in out Token Bucket as follows:

Table 4-3: Token Bucket Parameters

Parameters	**Definition**	**Specification**
Y	New packet arrival time	
$\overline{Y}'$	New mean packet arrival time	$\overline{Y}' = (1-\alpha)\times\overline{Y}+\alpha\times Y$
$\overline{Y}$	Mean Packet arrival time	$\overline{Y}=\overline{Y}'$
σ	Variance of arrival time	$\begin{cases}\overline{Y'}=(1-\alpha)\times\overline{Y}+\alpha\times Y\\ \overline{Y'^2}=(1-\alpha)\times\overline{Y^2}+\alpha\times Y^2\end{cases}$ $\sigma^2=\overline{Y'^2}-\overline{Y'}^2$
S	Packet size.	16000bits
D	Total delay requirement	200ms
r	Token arrival rate	$\frac{S}{\overline{Y}}$
r'	new token arrival rate	
Rmin	MIN token arrival rate	$\frac{S}{\overline{Y}}$
B	bucket size	$r\times\sigma$
b	remaining tokens	
11D	802.11e delay	Queue + MAC + Transmission delay
16TD	802.16 token bucket delay.	$0\sim\frac{S-b}{r}$
16DReq	802.16 delay requirement	D – 11D – 16TD
BW Req	802.16 bandwidth requirement.	$rf+\frac{b}{\left\lfloor\frac{16DReq}{f}\right\rfloor-1}$

4.2.2. Token Rate and Bucket Size Initialize

When the first connection arrivals SSQAP, it will initial the token rate r and bucket size B. We can get the arrival time Y and estimate the mean arrival time $\overline{Y}$ and variance of arrival time σ. The formula as follows:

$$r = \frac{S}{\overline{Y}} \tag{17}$$

$$B = r \times \sigma \tag{18}$$

4.2.3. Token Rate and Bucket Size Tuning

After the initialize, SSQAP must keep monitoring the arrival time for tuning the token rate r and bucket size B. The measure is as follows:

If $\begin{cases} \overline{Y}' \geq 1.5 \times \overline{Y} \\ \overline{Y}' \leq 0.5 \times \overline{Y} \end{cases}$, then reset token rate r and bucket size B.

If $b < B/2$, then we assume there are burst data recently and need to tune the token rate r and the bandwidth must be enough. We can estimate there are b tokens in bucket and the extra bandwidth requirement is $\frac{b}{\left\lfloor \frac{16DReq}{f} \right\rfloor - 1}$ referred to section 2.2. Thus, the total bandwidth is $rf + \frac{b}{\left\lfloor \frac{16DReq}{f} \right\rfloor - 1}$ (average + extra). We can tune the token rate r at next superframe. The relationship and the algorithm are as follows:

New average bandwidth = Old average bandwidth + Extra bandwidth

$$r'f = rf + \frac{b}{\left\lfloor \frac{16DReq}{f} \right\rfloor - 1}$$

Therefore, we could estimate the new *r'* .

$$r' = r + \frac{b}{\left(\left\lfloor \frac{16DReq}{f} \right\rfloor - 1 \right) * f}$$

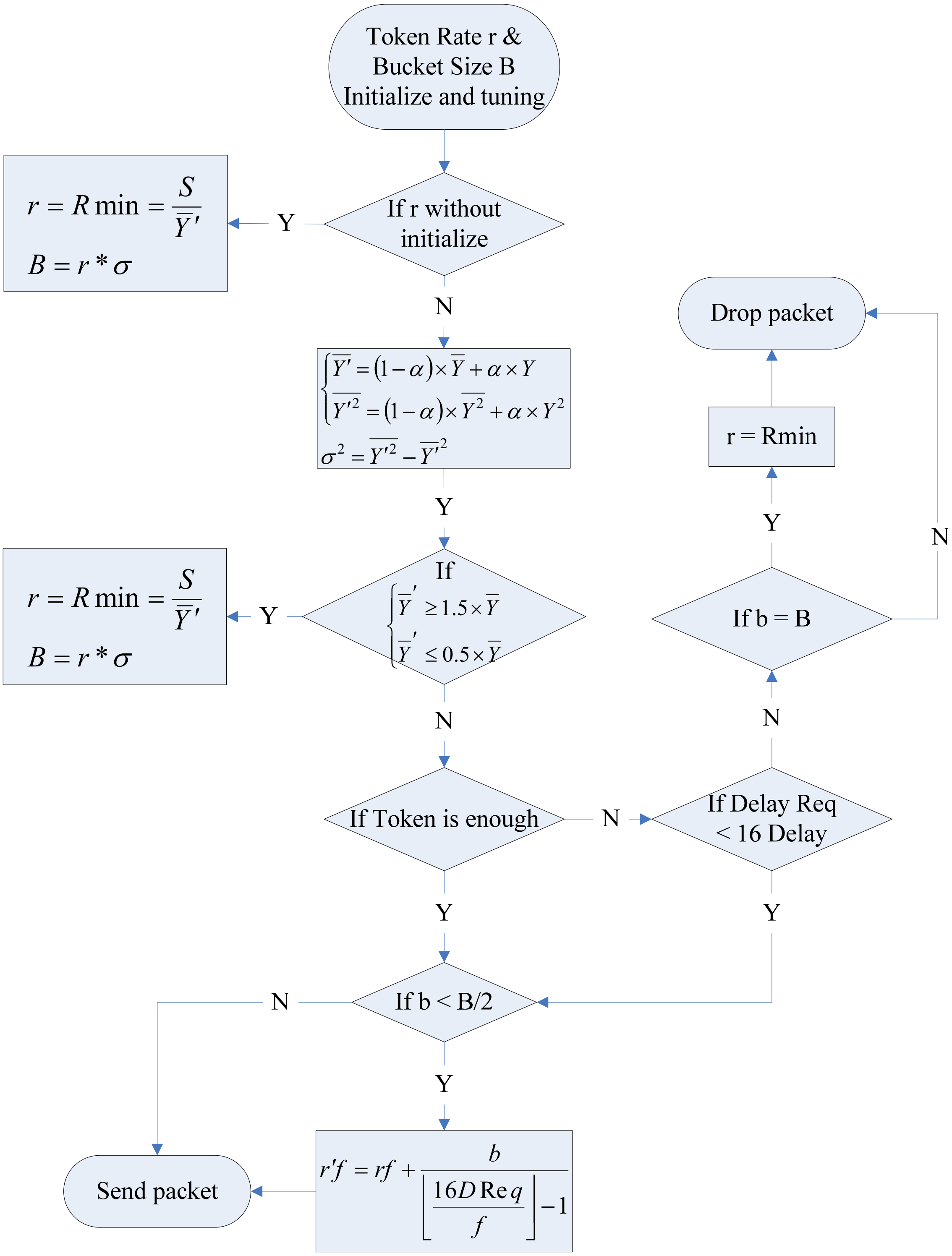

Figure 4-2: Transmission Time for COLL, SMALL, and LARGE.

4.3. Packet Drop Mechanism

Now, we can estimate the connection number for constraining the traffics arrival to SSQAP and automatic tuning token rate *r* for meeting bandwidth requirement. We can control the connection number for reaching QoS guarantee. Nevertheless, if the traffic loading is very heavy and leads the packet queue to be almost full then new packets drop rate is too high, that will result in the packet drop rate increase and throughput decrease. Thus, we use a brief packet drop mechanism to enhance the throughput.

If the loading is heavy and the packet queue is full, the new arrival packet will be dropped. However, the packet delay in front of queue still may be over the delay requirement. For this reason, when the forefront packet delay in queue reaches the delay requirement, we drop four packets together. Because the packet inter-arrival time (PIAT) of VI is about 42 ms and the delay requirement is 200 ms. We drop four packets can improve the probability of the packet which wants to transmit next time. Although, this way can not increase packet drop rate, but can enhance the throughput efficiently.