Rate Control Adaptation as a Bandits problem

M.S.Krishna Deepak, Guide: Mythili Vutkuru, IIT Bombay

May 3, 2015

1 Introduction

Rate adaptation is the determination of the optimal data transmission rate most appropriate for current wireless channel conditions. It consists of assessing channel conditions and accordingly adjusting the rate. Most widely used algorithm in wireless cards is the Minstrel algorithm.

In this report, we would first analyze the effect of look around rate in minstrel on various factors like throughput, delay, jitter. These factors have been observed by implementing nodes and wireless network in ns3, a network simulator tool.

2 Rate control and Bandits problem

We would also model the rate control problem as a bandits problem. Each bit rate in wireless can be thought of as an arm in a bandit problem. The probability of success when an arm is pulled can be related to the throughput achieved which we try to maximize.

3 Minstrel Algorithm

This is one of the most widely used rate control algorithm today. We maintain certain set of statistics like throughput, probability of success for every rate. Here, we try to choose the bit rate which provides maximum throughput. We update statistics once in every 0.1 second. EWMA is used in calculation of these statistics to make the algorithm dynamic.

In minstrel, there are two categories of transmission: normal transmission occurs 90 percent of the time and sampling transmission occurs for the remaining 10 percent of packets. During normal transmission the rate that achieves the highest expected throughput is used to send packets. While during look around(sampling transmission) a random rate is chosen to sample. As the channel situations change, the look around packets help to find out the new rate which achieves maximum throughput.

4 Experiments in ns3

Experiments are performed in ns3 to help in analysis of how changing look around rates affects the rate chosen, throughput achieved and some other statistics. Currently, minstrel algorithm sets this value at 10 percent.

4.1 Node setup in ns3

We have 2 nodes - access point(ap), station. Wifi Channel between the two nodes is setup using YansWifiChannelHelper. Physical layer is setup using YansWifiPhyHelper, Mac layer is setup using NqosWifiMacHelper at each node. We use wifiHelper to set the other parameters like Wifi standard, rate control algorithm being used. The standard used is 802.11a and the rate control algorithm used is minstrel. Both nodes are initially set at the origin. We use a BulkSendHelper and PacketSinkHelper at the application level to send packets in bulk. Link to the code is provided in references.

To see the effect of lookaround rate, we move the station node at a constant speed(which can be tuned) to a distance of 100 and bring it back at the same speed to the initial place. The distance of 100 is chosen because only then can we reach a situation where maximum throughput rate chosen by minstrel is also the lowest one in its list of choices. So, we try to see how quickly the minstrel algorithm adapts to such a dynamic situation and this depends on the variable 'LookAroundRate' which can be found in the minstrel code. In the following two sections, we perform experiments on this node setup.

4.2 Effect of Look Around rate on the rate chosen

Here we try to show how look around rate affects the rate chosen by minstrel. In the following graphs, speed is unchanged(set to 2). We have a point for every packet and the y-axis shows the rate at which it was sent. Look around rate is set to 3 in figure 1, increased to 10 in the second figure 2 and further increased to 30 in the third figure 3. By a time of 50 sec, station node has reached distance of 100 which is the farthest point.

We can see that lower look around rates tend to use max throughput rate without doing much look around. While higher look around rates tend to use other rates quite frequently. We can also see that since we are using low speed value, there is not much effect in increasing the look around rate. The algorithm is able to adapt to situation(able to find the max throughput rate quickly) even if look around rate is 3.

4.3 Effect of Look Around rate on other parameters

In following figures, each line corresponds to a speed value. At a chosen speed value, we increase the look around rate and plot various parameters on y-axis. Then, we increase the speed itself to get different plots. Parameters like throughput, average packet delay, standard deviation of the delay(jitter) and average retry rate have been plotted on y-axis.

Retry rate - Figure 4 shows this graph. There is a linear relationship between the look around rate and the retry rate. Increasing look around rate leads to high retry rates. The graph shows this property at all speeds.

When we are at farthest position from the access point, the best rate to send packets is the least one. Higher look around rates will lead to

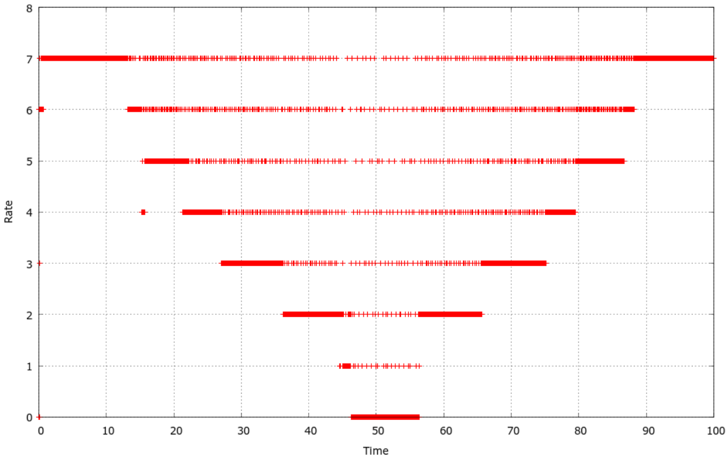

Figure 1: Look Around Rate - 3

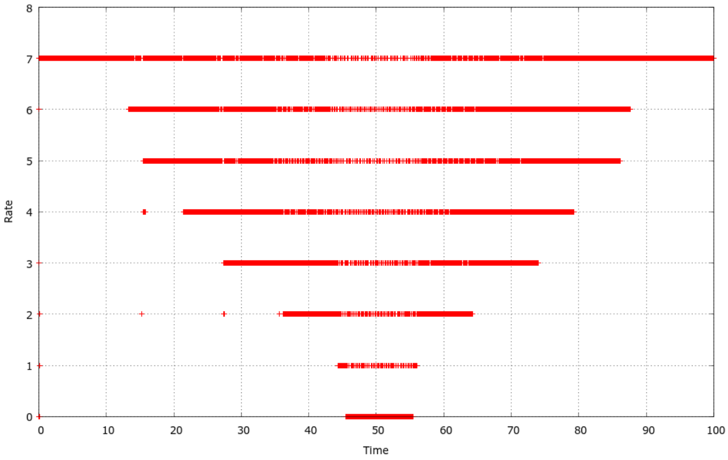

Figure 2: Look Around Rate - 10

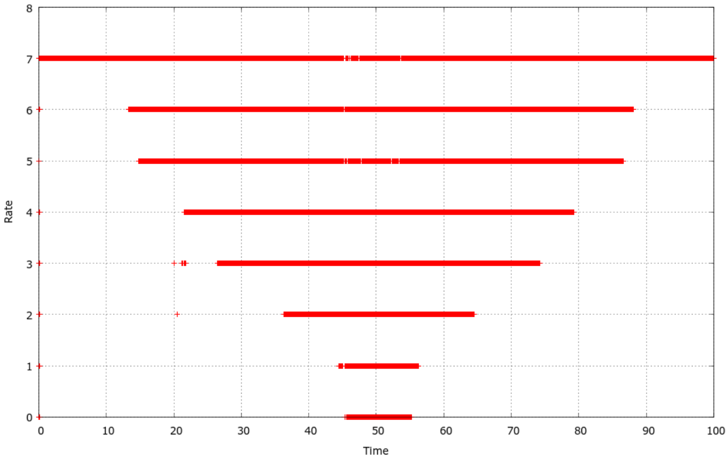

Figure 3: Look Around Rate - 30

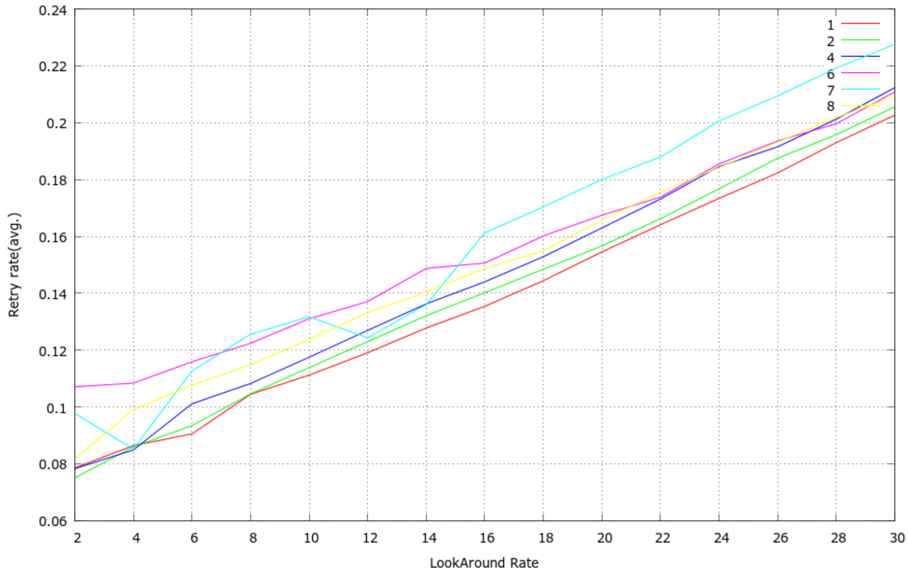

Figure 4: Retry rate vs Look Around Rate

exploring higher rates and since all the other rates are going to fail, high retry rates occur.

Throughput - Figure 5 shows this graph. y-axis shows the throughput obtained in Mbps. At lower speeds, increasing look around rate leads to a decrease in throughput because the channel is not too dynamic and a lower look around rate is enough to find the max throughput rate at any time.

At higher speeds, we can see the effect of look around rate. There is an initial increase in throughput with look around rates, after which throughput decreases. As the channel situations are changing quickly, lower look around rates are not good enough to find the maximum throughput rate. This explains the initial increase in throughput. After reaching a certain look around rate, this effect seems to be nullified by other factors like high retry rate.

Although the amount of changes in throughput is of the order of 1Mbps, we can clearly see the emerging property.

Delay - Figure 6 shows this graph. This graph shows the average packet delay in seconds on y-axis. This graph follows from the explanation provided for the throughput graph and retry rate graph. Generally, we expect the delay to increase with increasing look around rates because of high retry rates. At high speeds, delay values decrease initially, after which they increase. This is because we are not able to choose the maximum throughput rate at high speeds and low look around rates, leading to sub-optimal rate being selected as maximum throughput rate.

Standard Deviation - Figure 7 shows this graph. On y-axis we have the standard deviation of packet delay. Standard deviation increases with increas-

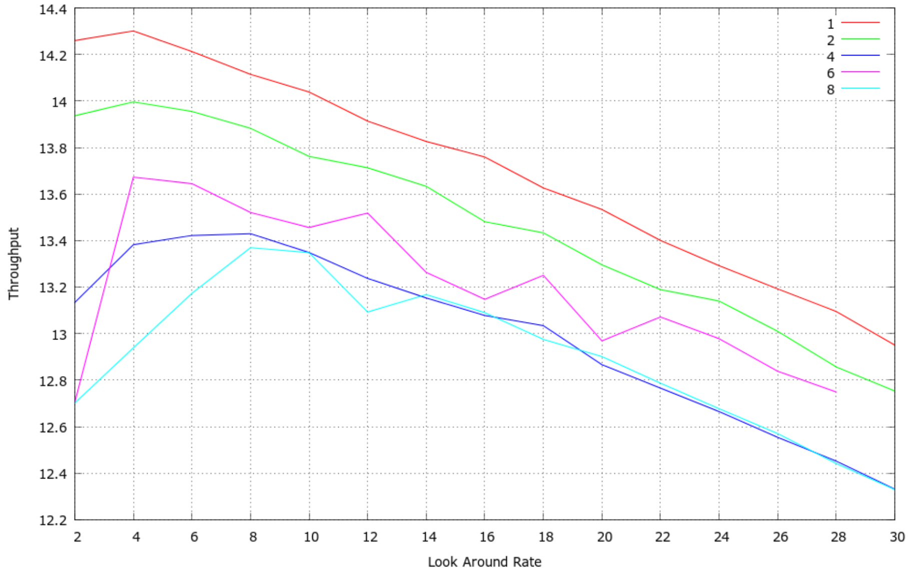

Figure 5: Throughput vs Look Around Rate

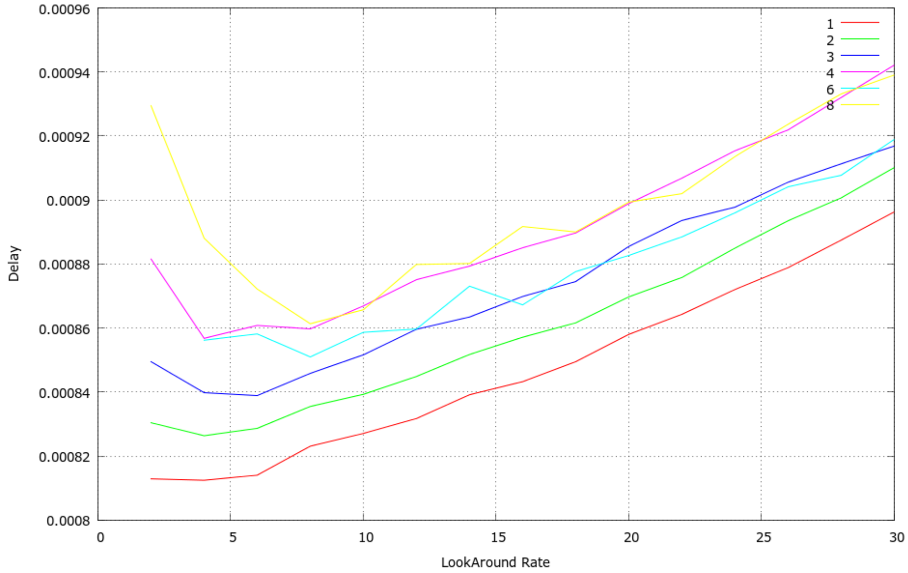

Figure 6: Average Packet Delay vs Look Around Rate

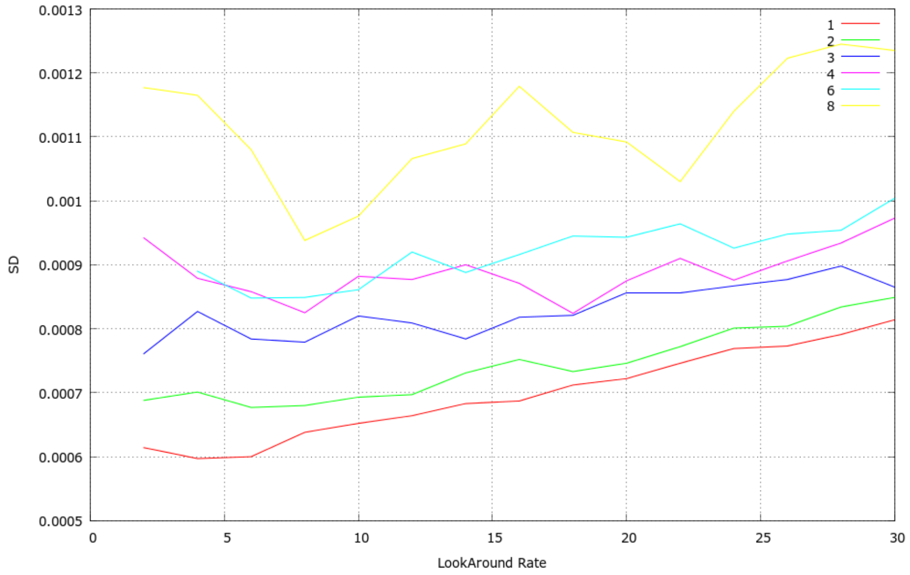

Figure 7: Standard Deviation in packet delay vs Look Around Rate

ing speeds as channel situations change dynamically leading to fluctuating packet delay rates. The effect of look around rate is not much here.

5 Formulations

5.1 Rate adaptation problem using Thompson Sampling

Let $Si(t)$ be the number of successes (reward = 1) and $Fi(t)$ be the number of failures (reward = 0) observed on pulling arm i at time t. The arm is determined by sampling from a Beta distribution of number of successes(Si) and failures(Fi) of that particular arm i. Let r is the result when arm i is chosen. If $r = 1$, then $Si = Si + 1$, else $Fi = Fi + 1$.

In the context of rate adaptation problem, each bit rate can be thought of as an arm. There are some important differences between the Thompson sampling and rate control algorithms.

- The measure of success when a bit rate is chosen should be proportional to the throughput it has achieved. So, a higher throughput should increase Si value by a higher amount than a lower throughput. Similarly, for every retry done at a particular rate i, Fi should be increased.

- As we have to give more weightage to a recent success than older ones, EWMA can be used to calculate new Si and Fi values.

$$Si(t+1) = \alpha * Si(t) + (1-\alpha) * x \qquad (1)$$

where x is proportional to throughput achieved. Similarly,

$$F_i(t+1) = \alpha * F_i(t) + (1-\alpha) * 1 \qquad (2)$$

- Older values of Si and Fi can be removed so as to allow our algorithm to adjust to dynamic situations.

5.2 Look Around Rate using Thompson Sampling

We can also model the look around rate in minstrel algorithm using Thompson sampling. We have a Beta distribution with two parameters α and β. Whenever we have a change in the maximum throughput, indicating that channel situation is changing, we increase α and if we don't have a change we increase β. We select a look around rate by sampling this beta distribution.

Hence, a static situation would mean increase in β leading to low look around rates, while a dynamic situation would mean increase in α leading to higher look around rates.

5.3 Rate adaptation problem using KL-UCB

Given n rates, choose each one of them for the first n times. After this we select the arm with the highest Upper Confidence Bound(ucb).
The equation for calculating ucb of rate i at time t is given by

$$ucb_i(t) = mean_i(t) + \sqrt{\frac{2 * ln(t)}{u_i(t)}} \tag{3}$$

$u_i(t)$ is the number of times arm i is chosen till now and $mean_i(t)$ is the average throughput obtained when this rate is selected.

- With upper confidence bounds, we make sure that each rate is chosen for some number of times.

- To calculate the average throughput for each rate, we can use probability of success as in minstrel. N_S is the number of packets transmitted successfully at the data rate and N_T is the total number of packets attempted at the data rate. α is the EWMA parameter.

$$R_S = N_S/N_T \tag{4}$$

$$P_{t+1} = R_S * (1 - \alpha) + P_t * \alpha \tag{5}$$

$$T = (P_t) * (B/t) \tag{6}$$

- The expected throughput T is the number of bytes B transferred in time t scaled by the probability of success.

6 Conclusions

- We have initially analyzed look around rate parameter of minstrel in great detail. We have observed that in situations which are dynamic, low look around rates are not desirable.

- Similarly, in situations which are static, we would want to be at lower look around rates to avoid sending unnecessary packets at other rates. Although the difference in throughput is not more than 1 Mbps, the property can be seen.

- To observe this in a better way, we would want a larger spectrum of bit rates available to choose from. Currently, ns3 does have newer standards like 802.11n which provide much larger spectrum of rates than 802.11a, but minstrel is not completely implemented in them.

- After this we have applied some bandit problem solutions to our context of wireless bit rate adaptation

7 References

1. https://github.com/mskd96/ns3 - Link to the code and statistics collected

2. http://linuxwireless.org/en/developers/Documentation/mac80211/RateControl/minstrel/ - Minstrel algorithm documentation

3. http://www.nsnam.org/docs/doxygen/minstrel-wifi-manager_8cc_source.html - Documentation of Minstrel code in ns3

4. http://arxiv.org/pdf/1102.2490v5.pdf - KL-UCB algorithm for bandits problem

5. http://arxiv.org/pdf/1102.2490v5.pdf - Thompson's algorithm for bandits problem