Effect of Time Delay in Network Traffic Control

Sthanunathan Ramakrishnan, Shivkumar Kalyanaraman, John Wen , Hitay Ozbay *†

September 22, 2000

Abstract

In this paper we consider the effect of timedelay in the performance of the AIMD algorithm, specifically queue length and fairness for both rate and window based sources. The queue bounds can be used to allocate buffers at bottlenecks so that flows can go through without any loss.We modify the AIMD algorithm by making the additive rate increase parameter proportional to the round trip time and the decrease parameter proportional to the power of round trip time which makes the algorithm fair even with heterogenous round trip times.

1 Introduction

The presence of time delays in computer networks has made the problem of analysing the performance characteristics of congestion control algorithms a very hard one. In this paper we study the effect of time-delay in the performance characteristics of the Additive Increase Multiplicative Decrease (AIMD) algorithm, like queue bound and fairness and suggest measures to improve fairness for both rate based and window based schemes.

In [3] the authors have derived various metrics for the single source case with constant RTT. In [4] [5], the authors show using a probabilistic model that AIMD can be thought of as trying to maximize a logarithmic utility function for a multiple bottleneck topology consisting of sources with heterogenous RTTs. Our paper differs from [3] in that we consider heterogenous RTTs and from [4] and [5] in that we consider a more accurate deterministic algorithm and present a simple graphical analysis to study AIMD's fairness. We then modify the algorithm by making the additive rate increase parameter proportional to the round trip time and the decrease parameter proportional to the power of round trip time to make it fairer. Floyd [2] has suggested a scheme which modifies the increase parameter in exactly the same manner but leaves the decrease parameter untouched, in the context of TCP congestion control.We show that the scheme will be fair only for schemes with small β (as in TCP where $\beta = .5$) and that the longer RTT source hogs a large share of the bandwidth once β is made large.

The main contributions of our paper are:

- Queue bounds for rate based and window based AIMD
- Simple graphical analysis of fairness of AIMD for both rate based and window based schemes
- A modified version of AIMD called Fair AIMD for both rate and window based AIMD

Section 2 describes the model used to analyse the AIMD algorithm, derivation of queue length bound and analysis of fairness of AIMD.. In Section 3, we derive our modified scheme Fair AIMD and a bound on its queue length and study its fairness properties. In section 4, we include the simulation results to confirm our analysis. Then we summarize our results and present conclusions in section ??.

2 Rate based AIMD

Let us first describe the AIMD algorithm briefly. The sources increase their rates linearly as long as there is no congestion and decrease their rates multiplicatively after the onset of congestion once every round trip time. Although there are various ways of detecting congestion, we consider the case in which the bottleneck declares congestion when the queue increases over a threshold and sets a bit in the packets to 1 thereby indicating that the bottleneck is congested In this Paper we neglect the granularity of data traffic and consider a fluid approximation to the traffic generated by sources [3].Consider the case of a single

*Sthanunathan Ramakrishnan,Shivkumar Kalyanaraman and John Wen are with the ECSE department, Rensselaer Polytechnic Institute, E-mail: sthanu@networks.ecse.rpi.edu, shivkuma@ecse.rpi.edu,wen@cat.rpi.edu

†Hitay Ozbay is with the Electrical Engineering Department,Ohio State University, E-mail:ozbay@ee.eng.ohio-state.edu

bottleneck with a constant capacity c fed by n sources with the round trip time of source i being τ_i.

The dynamics of the bottleneck queue $q(t)$is given by

$$\dot{q}(t) = \left\{ \begin{array}{ll} max(\sum_{i=1}^{n} r_i(t) - c, 0) & \text{q=0} \\ \sum_{i=1}^{n} r_i(t) - c & q > 0 \end{array} \right\} \quad (1)$$

Since the server is bottlenecked we expect $q > 0$ always and hence neglect the nonlinear term for $q = 0$ from our analysis.Let the increase constant be a and the decrease factor be β for all the sources. Let q_d be the desired queue length above which the bottleneck marks the packets. Let $r_i(t_{q_d} + \tau_i)$ be the rate at the beginning of the decrease phase, where t_{q_d} is the time at which the queue increases over q_d and $t_{q_d} + \tau_i$ is the time at which the source starts decreasing its rate. During the increase phase the rate increases by a every τ_i.During decrease phase the rates are multiplied by a factor β ($\beta < 1$) once every τ_i. So we can model the rates in continuous time as

$$\left\{ \begin{array}{ll} \dot{r}_i(t) = \frac{a}{\tau_i} & q(t-\tau_i) \leq q_d \\ r_i(t) = r_i(t_{q_d} + \tau_i)\beta^{\frac{t}{\tau_i}} & q(t-\tau_i) > q_d \end{array} \right.$$

2.1 Queue bound for rate based AIMD

Now we attempt to characterize an upper bound on the queue length.The queue increases and hits q_d at time say t_1.Let $t_2 = t_1 + \tau_{max}$ be the time by which all the sources know that congestion has set in and start cutting down the rates.Let t_3 be the time at which the queue reaches a maximum. Let $q(0)$ be the initial queue length.At this turning point of the queue $\sum_{i=1}^{n} r_i(0) = c$ where $r_i(0)$ is the rate of the i_{th} source at that instant.The equations governing the increase of rates and queue for $q(t) \leq_d$will be

$$r_i(t) = r_i(0) + \frac{at}{\tau_i}$$

$$\dot{q}(t) = \sum_{i=1}^{n} \frac{a}{\tau_i} t$$

Now we define τ_{eq} as $\frac{n}{\tau_{eq}} = \sum_{i=1}^{n} \frac{1}{\tau_i}$ So τ_{eq} can be interpreted to be the equivalent equal delay that all sources should have to get the same rate and queue variation at the bottleneck.

$$q(t) = q(0) + \frac{na}{2\tau_{eq}} t^2$$

$$= q_d \quad at \quad t = t_1$$

$$t_1 = \sqrt{\frac{2\tau_{eq}(q_d - q(0))}{na}}$$

If t_1 is larger $r_i(t)$ is larger and the queue takes a longer time to drain. So for worst case queue bound q(0)=0. So in this case

$$t_1 = \sqrt{\frac{2\tau_{eq} q_d}{na}}$$

and

$$r_i(t_1) = \frac{\sqrt{\frac{2a\tau_{eq} q_d}{n}}}{\tau_i} + r_i(0)$$

$$\sum_{i=1}^{n} r_i(t_1) = \sqrt{\frac{2naq_d}{\tau_{eq}}} + c$$

Between times t_1 and t_2 all sources will realise the onset of congestion and cut down their rates.It can be easily seen that at t_2

$$r_i(t_2) = (r_i(t_1) + a)\beta^{(}\frac{\tau_{max} - \tau_i}{\tau_i})$$

But the expression for queue becomes very unwieldy and since we are anyway interested only in an upper bound we can assume that all the sources increase till t_2 and then decrease their rates.Therefore

$$r_i(t_2) = (r_i(t_1) + \frac{\tau_{max}}{\tau_i} a$$

$$\sum_{i=1}^{n} r_i(t_2) = c + \sqrt{\frac{2naq_d}{\tau_{eq}}} + \frac{\tau_{max}}{\tau_{eq}} na$$

$$q(t_2) = q_d + \frac{na\tau_{max}}{2\tau_{eq}}(2t_1 + \tau_{max})$$

After the sources start cutting down the rates $\dot{q}$ decreases and becomes 0 at t_3.At this instant the queue reaches a maximum.It is easy to see that

$$\sum_{i=1}^{n} r_i(t_2)\beta^{\frac{t_3 - t_2}{\tau_i}} = c$$

It is not possible to get a closed form solution for this equation. But to get an upper bound we can assume that τ_i is replaced by τ_{max}. This means that the sources cut down their rates only every τ_{max} and this should obviously lead to a higher value for t_3 and q_{max}. We can see that this assumption will be more valid when the rate of the source with max RTT is maximum or when $r_i(0) = c$ ($i \ni \tau_i = \tau max$).

$$t_3 - t_2 = \frac{\tau_{max}}{\ln \beta} \ln \frac{c}{c + \sqrt{\frac{2naq_d}{\tau_{eq}}} + \frac{\tau_{max}}{\tau_{eq}} na}$$

$$\begin{aligned} q_{max} = q(t_3) \quad &= \quad \sum_{i=1}^{n} \frac{r_i(t_2)\tau_i}{\ln \beta}(-1 + \beta^{\frac{t_3 - t_2}{\tau_i}}) \\ &\quad -c(t_3 - t_2) + q_d + \frac{na\tau_{max}}{2\tau_{eq}}(2t_1 + \tau_{max}) \end{aligned}$$

We know all the parameters since $r_i(0) = c$ ($i \ni \tau_i = \tau max$) and $r_i(0) = 0$ otherwise for maximum queue overshoot. We can simplify this expression by setting $\tau_i = \tau_{max}$ thereby weakening our upperbound further.Then

$$\begin{aligned} q_{max} &= -\frac{\tau_{max}}{\ln\beta}(\sqrt{\frac{2anq_d}{\tau_{eq}}} + \frac{na\tau_{max}}{\tau_{eq}}) \\ &-\frac{c\tau_{max}}{\ln\beta}\ln\frac{c}{c+\sqrt{\frac{2naq_d}{\tau_{eq}}}+\frac{\tau_{max}}{\tau_{eq}}na} \\ &+q_d + \frac{na\tau_{max}}{2\tau_{eq}}(2t_1+\tau_{max}) \end{aligned}$$

We have plotted the real queue obtained from Matlab simulations with initial conditions as discussed in figure 2 and the graph shows that the bound is reasonable.From the expressions for q_{max} we can see that our approximations are equivalent to all the sources having a timedelay of τ_{eq} during the increase phase and a time delay of τ_{max} during the decrease phase. This expression for q_max can be used to allocate buffers at bottlenecks as the network managers typically know the maximum RTT and the average number of sources.

2.2 Queue bound for window based AIMD

In window based AIMD the sources increase or decrease their window sizes once every RTT. The model that we consider is closer to TCP-ECN as we do not consider losses and timeouts that are characteristic of other versions of TCP. RTT has two components viz propagation and transmission delay(nominal delay) and queueing delay.In rate based sources the increase or decrease is done once in every estimated RTT which is obtained by averaging individual RTT values over time.Since the queue oscillates around q_d the queueing delay component in the RTT estimate will be a constant equal to that for q_d.But window based sources increase or decrease their windows only after the acks for the packets sent in previous rounds have been received. This causes the equivalent input rate of the source to depend on the queueing delay. If we consider different nominal delays as well as queuing delays in the analysis of window based AIMD then we cannot obtain closed form solutions. So we step back and assume equal nominal time delays for all sources. We also consider a discrete time model as a continous time model becomes mathematically intractable.

Let $q(k)$ be the queue at the beginning of the kth RTT and let $w_i(t)$ be the window size of the i_{th} source during the k_{th} RTT.Let T be the nominal delay of all sources and let a be the increase parameter and β the decrease parameter.Let k_1 be the round at which $q(k) = q_d$.Let at round $k = k_1 + k_2$, the queue reach the maximum.The equations governing the system are:

$$q(k+1) = q(k) + \sum_{i=1}^{n} w(k) - cRTT(k)$$

$$RTT(k) = \tau + \frac{q(k)+q(k+1)}{2c}$$

In the first equation if we divide throughout by $RTT(k)$ and compare it with the equation (1) for rate based AIMD, we note that $\frac{w(k)}{RTT(k)}$ can be thought of as an equivalent rate. we will use this property in deriving fairness properties of window based AIMD. The second equation may seem non causal but in reality its not as RTT(k) can be known at the source only in the $(k+1)_{th}$ round.The packets sent out in round k face a queue that changes from $q(k)$ to $q(k+1)$ and hence we model the average queueing delay during that round as due to the average of the two queue lengths. We will soon see that the queue increases almost linearly thereby making our assumption almost exact.The dynamics of the window size is given by

$$w_i(k+1) = w_i(k) + a \quad q(k) <= q_d$$

$$w_i(k+1) = \beta w_(k) \quad q(k) > q_d$$

Lets suppose that queue starts increasing from k=0. For worst case queue length we have to choose q(0)=0 and the argument is similar to that for rate based AIMD.Since at index 0 the queue starts increasing $\sum_{i=1}^{n} w_i(0) = c$. Substituting for $RTT(k)$ and solving the difference equations for $w_i(k)$ and q(k) we get,

$$q(k) = na(k - .5 + \frac{1}{23^k})$$

$$= q_d \ mboxwhen \ k = k_1$$

Therefore $k1 \approx \frac{q_d}{na}$

$$\sum_{i=1}^{n} w(k_1) = cT + na(k_1+1)$$

$$= cT + q_d + na$$

Since the queue is affected by only the sum of the rates we set W(k)=$\sum_{i=1}^{n} w(k)$. Now solving for the queue during the decrease phase we get

$$q_k = \frac{1}{3}q_d + \frac{2}{3}W(k_1)\beta^{k-k_1}\frac{1-\frac{1}{3\beta^{k-K_1}}}{1-\frac{1}{3\beta}} - (1-\frac{1}{3^{k-k_1}})cT$$

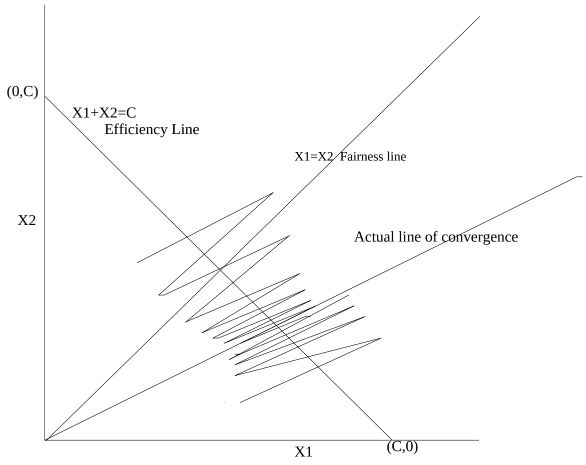

Figure 1: Convergence to Unfair steady state

Differentiating the above equation and setting it to zero at $k = k_1 + k_2$we get

$$k_2 = \log_{3\beta}((-\log_\beta 3)(1 - \frac{(3\beta - 1)cT}{2\beta W(k_1)}))$$

Substituting this in the equation for q we get an expression for q which can be approximated as

$$q_{max} \approx \frac{1}{3}q_d + \frac{2\beta}{3\beta - 1}W(k_1)\beta_2^k - cT$$

Now that we have derived the expressions for the queue bounds we observe a few differences from the rate based case.The queue increases linearly as opposed to a quadratic increase for rate based sources.For values of c=10Mbps, T=100ms, n=12, a=512 bytes and $\beta = .9$ we get k=1.72. This means that within two RTTs the queue starts decreasing.Therefore we see that in the window based scheme there is a smoother variation of the queue due to the less aggressive control action.

2.3 Fairness Analysis of rate based AIMD

In this section we do a graphical analysis,very similar to a phase plot, of the fairness of AIMD with two sources following the procedure, adopted in [1].

Let the timedelays of the sources be τ_1 and $n\tau_1$ and let x_1 and x_2 be the maximum rates of the two sources respectively in one cycle of increase and decrease.During the increase phase the rates vary along a straight line of slope $\frac{1}{n}$ as shown in figure 1.During the decrease phase the rates change from (x_1, x_2) to $(\beta^{nk}x_1, \beta^k x_2)$ where k is the number of decreases for source 2.For fairness to improve we want the phase plot to move closer to the 45 degree equi-fairness line after every cycle. This means that for the region above the 45 deg line the slope during decrease should be greater than the slope during increase and vice-versa for the region below the 45 deg line.So the condition for fairness to improve are

$$\frac{x_2(1-\beta^k)}{x_1(1-\beta^{nk})} > \frac{1}{n} \quad given \quad x_2 > x_1$$

which implies

$$x_2 > \frac{x_1(1-\beta^{nk})}{n(1-\beta^k)} \quad given \quad x_2 > x_1$$

which is always true.

$$\frac{x_2(1-\beta^k)}{x_1(1-\beta^{nk})} < \frac{1}{n} \quad given \quad x_2 < x_1$$

implies

$$x_2 < \frac{x_1(1-\beta^{nk})}{n(1-\beta^k)} \quad given \quad x_2 < x_1$$

which is not true always. So fairness is not achieved and we can see that instead of converging to the $x_1 = x_2$ line the sources converge to the line $x_2 = \frac{x_1(1-\beta^{nk})}{n(1-\beta^k)}$.This shows that AIMD is unfair and that the unfairness increases as beta decreases and as n increases. Figure 1 illustrates this convergence for two initial conditions, one with x1>x2 and the other with x1<x2.A phase plot of the rates is in fig 4 and the simulation results are discussed in section 4 In this derivation we have assumed that the sources get consistent information i.e. at any point in time all the sources are either in increase mode or in decrease mode. This is a reasonable assumption as long as the ratio of the RTTs is not too large and as long as the length of the congestion epoch is much larger than the RTTs which will be true if the βs are not too small.Otherwise the congestion epoch might be over before the source with the longer RTT starts decreasing and our analysis fails.

2.4 Fairness Analysis of window based AIMD

Fairness means that the number of packets sent in a given time is the same for all sources. So the rates of the sources have to be equal to ensure fairness.We saw that the window can be replaced by an equivalent rate equal to the window size divided by the RTT in section 2.2. This means that the equivalent rate increase parameter is inversely proportional to the RTT and the decrease parameter remains unchanged.

Let the timedelays of the sources be τ_1 and $n\tau_1$ and let x_1 and x_2 be the maximum rates of the two sources respectively in one cycle of increase and decrease.The slope during increase will be $\frac{1}{n^2}$ and we can show by repeating the previous arguments that at steady state the system converges to

$$x_2 = \frac{x_1(1-\beta^{nk})}{n^2(1-\beta^k)}$$

This line is farther away from $x_1 = x_2$ than the one obtained for rate based AIMD. This shows that fairness is worse for window based sources.

3 Fair AIMD

An interesting question that arises is whether we can design a decentralized control algorithm that can drive the system to a fair operating region. We know from [1] that AIMD is fair if the rates increase by equal amounts and decrease by the same factor in a given time interval.This can be seen to be true from the graphical analysis by setting n=1 and observing that at steady state $x_1 = x_2$. We know that

$$\begin{cases} \dot{r}_i(t) = \frac{a}{\tau_i} & q(t-\tau_i) \leq q_d \\ r_i(t) = r_i(0)\beta^{\frac{t}{\tau_i}} & q(t-\tau_i) > q_d \end{cases}$$

So to achieve equal rate increase we can replace parameter a by $\frac{a\tau_i}{T}$ and to achieve the same factor of decrease we can replace β by $\beta^{\frac{\tau_i}{T}}$ for each source. Then the equations become

$$\begin{cases} \dot{r}_i(t) = \frac{a}{T} & q(t-\tau_i) \leq q_d \\ r_i(t) = r_i(0)\beta^{\frac{t}{T}} & q(t-\tau_i) > q_d \end{cases}$$

Thus the system is transformed to an equivalent system where all the sources have a time delay equal to T. Ofcourse its not entirely equivalent because when q increases over q_d or decreases below q_d upto a time equal to τ_{max} some sources are increasing while others are decreasing.But as long as the RTTs are not too widely different and as long as the length of the congestion epoch is much greater than the τ_{max} we expect the effect of this to cancel out and not affect the system performance significantly.

We can do a similar modification to window based sources by making the increase parameter $a \propto \tau^2$ and replacing the decrease parameter β by $\beta^{\frac{\tau}{T}}$. This will cause the system to become fair provided we neglect asynchronousness and inconsistent feedback.

3.1 Queue length bound

Notice in our analysis in section 2.1 that if all $\tau_i = \tau$ then our queue bound is exact.But we saw that fair AIMD behaves very much like the constant RTT case in normal AIMD except when the system is in the transition regions.But we could bound the behaviour in the transition regions by assuming that all sources increase until an extra time of τ_{max} elapses just as in section 2.1. Then we can calculate the queue bound for Fair AIMD essentially following the same procedure as before and derive

$$\begin{aligned} q_{max} &= -\frac{T}{\ln\beta}\left(\sqrt{\frac{2naq_d}{T}} + \frac{na\tau_{max}}{\tau_{eq}}\right) \\ &\quad -\frac{cT}{\ln\beta}\ln\frac{c}{c+\sqrt{\frac{2naq_d}{T}}} \\ &\quad +\frac{\tau_{max}}{\tau_{eq}na} + q_d + \frac{na\tau_{max}}{2T}\left(2\sqrt{\frac{2q_dT}{na}} + \tau_{max}\right) \end{aligned} \quad (2)$$

This queue bound is a pretty tight bound as compared to that for normal AIMD.Also the bound depends on T which is a parameter that can be tuned. A network manager could theoretically speaking set T such that q_{max} is minimised given all other parameters and thus minimise buffer overflows in the network. But typically the number of sources n is variable and so it may not be very useful. We also observe that q_{max} increases with T. So it will not be a bad choice to set $T = \tau_{min}$. But it may not be possible to know τ_{min} and hence we may have to set T somewhat arbitrarily.For sources with large RTTs the control action in fair AIMD is much more drastic as compared to normal AIMD. Also the feedback received from the bottleneck over one RTT may not be all equally important.So to minimise wrong increase or decrease we may want to take only the feedback in say the second half of the RTT.But this is difficult to do as a source doesnt know if its RTT is "large" or "small". If we can find a satisfactory way of doing this then we can increase the range of RTTs over which the system operates satisfactorily when deployed in the real Internet.

3.2 Fairness properties

Consider a scenario as in section 2.3. The slope during increase will be 1 and that during decrease would be $\frac{x_2}{x_1}$ and so we can see easily that at steady state the plot reaches the line $x_2 = x_1$.In the general n source case we can take recourse to the proof in [1] as each source behaves as if it had an RTT=τ.The simulation results are presented in section 4 . At this stage its pertinent to comment on the modification to AIMD suggested by S.Floyd in [2] where the additive rate increase parameter (and in the window case, the equivalent rate increase parameter) is made proportional to RTT while the decrease parameter is left unchanged. By a similar graphical analysis we can show that at steady state the system converges to the line

$$x_2 = \frac{1-\beta^{nk}}{1-\beta^{k}}\ x_1$$

which means that at steady state the longer RTT source will hog a large share of the bandwidth. The reason why Floyd's method works for TCP (as shown by simulations) is that in TCP $\beta = .5$ and this causes the congestion to be relieved almost always within one RTT. So our analysis doesnt really hold. But if this scheme is applied in Dec-bit schemes or TCP-ECN (where β is high) then the system will become unfair with the longer RTT source hogging a lot of the bandwidth.

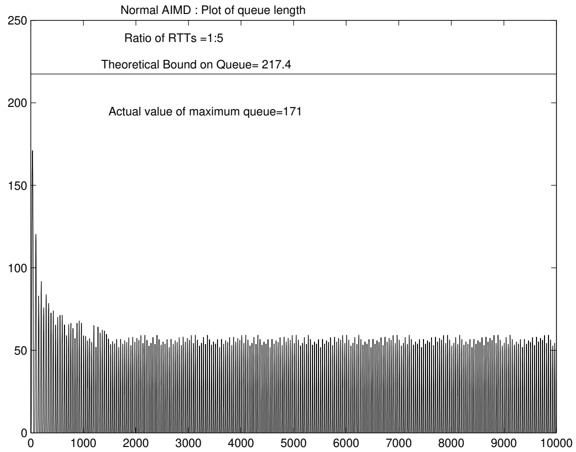

Figure 2: Queue variation in normal AIMD

4 Simulations

We ran matlab simulations for rate based normal AIMD and the modified fair AIMD to verify the results. The simulation set up had two sources with RTTs of 1 and 5 respectively. The capacity of the bottleneck=200 and the parameter a=1 and β=.98. For fair AIMD we set T=1. The queue length and the derived queue bound are plotted for AIMD in figure 2 and for fair AIMD in figure 3. We see that the results match The bound is much tighter in fair AIMD as predicted. We also see that the queue length variation is much more abrupt in fair AIMD than in normal AIMD. This because the sources behave as if their time-delays are all equal to 1 (we have set T=1). So the situation is similar to the case in normal AIMD with both sources having RTT=1.Obviously this means a faster rate of increase as well as decrease and hence the sharp variation in queue.

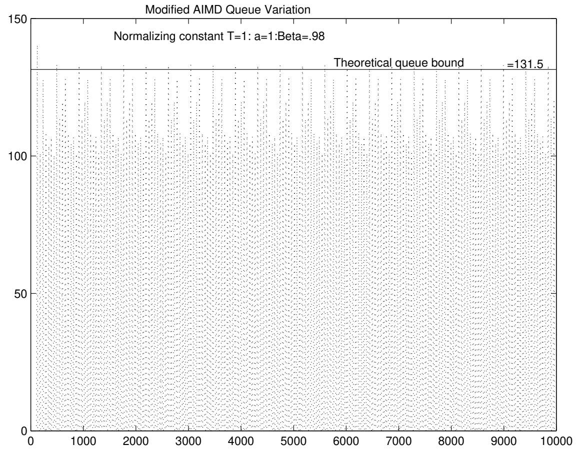

Figure 3: Queue variation in fair AIMD

The phase plot of the source rates for AIMD is in figure 4 and for fair AIMD in figure 5.For fair AIMD we drew the theoretically calculated steady state line by getting the average number of decreases that each source does and using it in the equation for the steady state line. We see that the source rates oscillate around the theoretically derived lines. An interesting observation is that the phase plot of AIMD has less oscillation while that of fair AIMD has larger oscillations.This is because the sources are not synchronised always and since the control action is more aggressive in fair AIMD the oscillation due to this asynchronous nature of rate updation is larger.

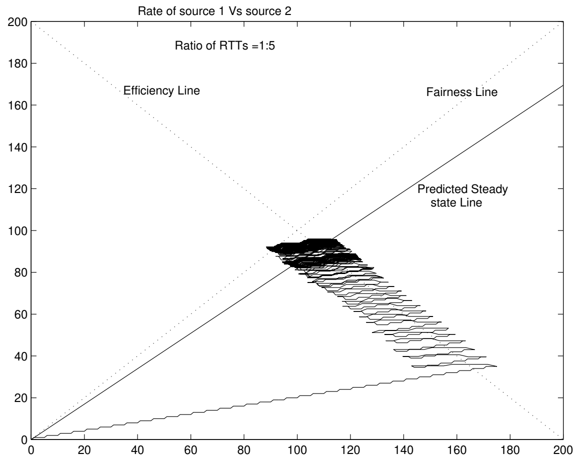

Figure 4: Phase Plot of normal AIMD

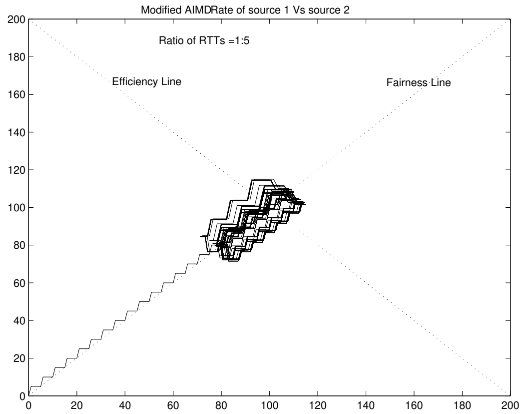

Figure 5: Phase Plot of Fair AIMD

5 Conclusion

In summary we have derived a bound on the queue length for rate based and window based AIMD. We also discuss the unfairness of AIMD with heterogenous time delays and suggest a modification called Fair AIMD to rectify the unfairness.Its interesting to note that there seems to be an iverse relationship between amount of oscillation and the fairness of the system.It would be interesting to explore these ideas further using a control theoretic framework.

References

[1] D-M Chiu and R.Jain. *Analysis of the Increase Decrease Algorithms for Congestion Avoidance in Computer Networks*, Journal of Computer Networks and ISDN systems,17(1)1989

[2] Floyd, S. *Connections with Multiple Congested Gateways in Packet-Switched Networks* , Computer Communication Review, Vol.21, No.5, October 1991, p. 30-47.

[3] J.C.Bolot, A.U.Shankar *Analysis of a fluid approximation to flow control dynamcics,* Proc.IEEE Infocom '92, Florence, Italy May 1992

[4] M.Vojnovic,J.L.Boudec and C.Boutremans. *Global fairness of additive-increasemultiplicative-decrease with heterogenous round trip times,* IEEE Infocom 2000, Tel-Aviv Israel, March 2000

[5] P. Hurley, J. Y. Le Boudec, P. Thiran. *A Note on the Fairness of Additive Increase and Multiplicative Decrease*, , Proceedings of ITC-16, Edinburgh, June 1999.