SIMULATION STDUDIES AND PERFORMANCE EVALUATION OF BACKBONE NETWORK TRAFFIC AGGREGATION

M.S. THESIS

BY

MEHER VINAY DHERAM

ADVISOR: PROFESSOR MING YU

Submitted in partial fulfillment of the requirements for
the degree of Master of Science in Electrical Engineering
in the Graduate school of
Binghamton University
State University of New York
2006

Accepted in partial fulfillment of the requirements for
the degree of Masters in Electrical Engineering in English
in the Graduate School of
Binghamton University
State University of New York
2006

August 18th 2006

Ming Yu, Department of Electrical Engineering, Binghamton University

Qing Wu, Department of Electrical Engineering, Binghamton University

Qinru Qiu, Department of Electrical Engineering, Binghamton University

ABSTRACT

In this thesis, we study the backbone network traffic aggregation problem by extensive simulations based on the method of clustering the representative time constants (RTCs) of Markov modulated Poisson processes (MMPP) models.

It is found that the decaying time constants of the aggregated traffic process are the products of the eigenvalues of the transition matrix of the individual traffic. If the time constants are well clustered around some RTCs, the corresponding states can be merged in the state space. In the worst case, if the time constants are uniformly distributed over the log-scale, we prove that there exist a minimum number of states that can approximate the traffic aggregation.

In this thesis, we design several numerical and simulation examples to demonstrate how to aggregate traffic described by MMPP models and how to reduce the order of the models while approximate the network performance, such as queue length distribution, mean delay, and packet loss ratio.

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge the enthusiastic supervision of Dr. Ming Yu during this thesis work. I would also like to thank the post graduate students Aniket Malvankar for the numerous stimulating, brainstorming discussions and general advice. I would like to thank the graduate students Shrinivas Kulkarni and Dennis Fernandes for their help in solving the technical problems.

Finally, I am forever indebted to my parents for their endless patience and encouragement when it was most required.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The diversity of the packet generation mechanism, the inherently bursty nature of data communications, and the various stages of switching the packets have to go through, often make the exact traffic modeling mathematically intractable. The major difficulties are due to the lack of appropriate description of the traffic aggregation processes and the large dimension of the model sizes.

In this Thesis, the main purpose is to develop an efficient traffic aggregation technique based on MMPP models. It enables us to gain analytical insight to practical traffic modeling, to which the existing methods could not be applied because of the state-space explosion problem.

Recently, the traffic measurements over high-speed data networks have shown that the traffic exhibits variability over many time scales. Especially, the second order properties of counting process seemingly display long-range dependence (LRD) and self-similarity. Therefore, it is natural to model the LRD over several time scales by fitting the covariance function of the two-state MMPP processes to that of the traffic measurements [3]. Here, one two-state MMPP is insufficient to fit the second order properties. In [4], the authors propose a special Markovian arrival processes (MAP) or MMPP structure and approximate the variance of arrival process at identified time scales by using wavelet

transform of finite sequences. But as the authors realized, Only capturing the first- and second-order characteristics of counts are known to be insufficient when attempting to predict queueing behavior.

In [5], it has been found that the asymptotic behavior of an infinite queue fed with different arrival processes that all exhibit the LRD property can be vastly different. Thus, it is also insufficient to consider only the correlation structure of the input process for accurate performance prediction. It has been found that the marginal distribution and the finite range of time scales of the arrival process must be also considered, as well as the correlation structure. Especially, the time scale associated with a queueing system is a function of the maximum buffer size. Thus, for finite buffer queues, the impact on performance of the correlation in the arrival process becomes nil beyond a time scale which was called correlation horizon (CH). Also, the loss rate depends in a crucial way on the marginal distribution of the fluid arrival process.

A direct impact of the discovery of CH is that more traditional traffic models such as MMPP can still be used to model traffic exhibiting LRD within the interest of time scales, if we can also fit the marginal distribution of the counting process. In [6], the authors propose a method to use L two-state MMPP's to the empirical auto-covariance with L time scales and use one M-state MMPP to fit the marginal distribution. Basically, this is a numerical fitting method. The total number of state used in the model is still very large when matrix exponential operations are involved. Clearly, the state-space explosion becomes the main hindrance to numerical solutions to the queueing model with

multiplexed MMPP arrivals [1].

A practical and more efficient way of modeling would be to use one multi-state MMPP for a single traffic stream and develop approximate algorithm to reduce the state space of the aggregated traffic. In [1], the authors developed an efficient algorithm to fit an MMPP model to trace data. The rates were chosen according to a predetermined spacing parameter ‘a’. The number of states corresponding to the number of rates chosen depends only on the largest and smallest values of the time series.

Related to the traffic aggregation problem, the following questions remain unanswered. How many states are needed for an approximate MMPP model of the aggregated traffic? If we know the number of states for the aggregation model, what are those most important or representative states that the model has to capture? If we just choose the maximum and minimum rates to decide the rest of rates to be modeled, we may miss the most representative ones. Also for a general model reduction, the rate limit is unknown, because we don't know which rates are more representative than the others. In this thesis work, we will try to answer these questions.

2. AGGREGATION BASSED ON MMPP MODELS

In this section, we assume that the network traffic is adequately modeled by MMPP.

2.1 Representative Time Constants

Mathematically, a discrete-time Markov chain (X, J) = {(X_k, J_k), k = 0, 1, … } with state space N_0 x S_0 is a discrete MMPP if and only if for k = 0, 1, …,

$$P(X_{k+1} = l, J_{k+1} = j | (X_k = m, J_k = i) = \begin{cases} 0 & \text{if } m < l \\ p_{ij} e^{-\lambda_i} \frac{-\lambda_i^{l-m}}{(l-m)!} & \text{if } m \geq l \end{cases}, \quad (1)$$

for all l, m $\in$ N_0 and i, j $\in$ S_0, with parameters Λ = diag(λ_i) and $P = (p_{ij})$, where, λ_i is the Poisson arrival rate to the state i of the Markov chain; p_{ij} is the transition probability between the states *i* and *j* of the Markov chain. We use the notation MMPP(P, Λ) to represent (X, J), where P and Λ are assumed to be time-invariant matrix of dimension n x n, where n is the number of the states of the Markov chain.

The irreducible stochastic matrix P has stationary probability vector π,

$$\pi P = \pi , \pi e = 1, \tag{2}$$

Where $e = (1,1, \ldots, 1)^{T}$ is a column vector of ones with length *m*. The mean arrival rate vector is

$$\lambda = (\lambda_1, \lambda_2 \ldots, \lambda_n)^{T} \tag{3}$$

It is well-known that the superposition of two MMPP processes, $MMPP_1(P_1, \Lambda_1)$ and $MMPP_2(P_2, \Lambda_2)$, is also an MMPP process with parameters:

$$\begin{aligned} P &= P_1 \otimes P_2 \\ \Lambda &= \Lambda_1 \oplus \Lambda_2 \end{aligned} \tag{4}$$

where $\otimes$ and $\oplus$ denote the Kronecker product and the Kronecker sum, respectively [7]. If $MMPP_1$ and $MMPP_2$ are of orders n_1 and n_2, respectively, then the order of the superposition process is $n = n_1 \times n_2$. This is the so-called state-space explosion problem.

As pointed out in [1], it is much easier to obtain traffic measurements as counts over discrete intervals than interarrival times of individual packets. But the continuous-time MMPP is better suited as an arrival process in extant algorithms and software queueing models than is the discrete-time MMPP. Therefore, it is preferred to convert the discrete-time MMPP to a continuous MMPP when conducting queueing analysis, i. e., to find an in infinitesimal generator $Q = (q_{ij})$ that corresponds to the discrete transition matrix P of the discrete MMPP.

One way is to choose

$$Q = P - I \tag{5}$$

In this way, both processes will have the same mean sojourn time in every state i, probability of any given sequence of states, and steady-state distribution. The stationary vector of the Markov chain can be also obtained from

$$\pi Q = 0, \qquad \pi e = 1. \tag{6}$$

Let N_t be the number of arrivals in (0, t]. The moment generating function of N_t is given in [7]:

$$\phi(z, t) = \pi \exp\{[Q + (z - 1)\Lambda]\, t\}\, e. \tag{7}$$

For the time-stationary version of the MMPP,

$$E(N_t) = \pi\lambda t, \tag{8}$$

$$V(N_t) = \pi\lambda t + 2t[(\pi\lambda)^2 - \pi\Lambda(Q + e\pi)^{-1}\lambda] + 2\pi\Lambda(e^{Qt} - I)(Q + e\pi)^{-2}\lambda. \tag{9}$$

For a two-state MMPP, there exist many methods to approximate both the mean and variance functions of the MMPP process [3], [6], [8]. For a general MMPP, it will be too complicated to fit the variance-time curve. Even if we found a good fit to the variance-time curve, it still may not help us in accurately predict network performance, as we discussed before.

It can be seen from Eqn. (9) that the decaying time constants of the variance function are determined by e^{Qt}. Moreover, in terms of the definition of Q in Eqn. (5), by combining Eqn. (4), we have

$$e^{Qt} = e^{[P_1 \otimes P_2]t} e^{-t} \tag{10}$$

where we use $e^{-It} = e^{-t} I$.

To simplify the discussion here, we first assume that P_i, $i = 1, 2$, have distinct eigenvalues $r_{i1}, r_{i2}, \ldots, r_{in}$ and eigenvectors $Y_{i1}, Y_{i2} \ldots, Y_{in}$, respectively. Thus, $(P_i - r_{ij} I_n)Y_{ij} = 0$, $i = 1, 2$, $j = 1, 2, \ldots, n_i$. We define a transformation matrix $\Gamma = (Y_{i1}, Y_{j2} \ldots, Y_{in})$. Then we have

$$\Gamma_i^{-1} P_i \Gamma_i = \text{diag}\,(r_{i1}, r_{i2}, , \ldots, r_{in}). \tag{11}$$

And

$$e^{Pit} = \Gamma_i \,\text{diag}\,(e^{r_{i1}t}, e^{r_{i2}t}, \ldots, e^{r_{in}t})\, \Gamma_i^{-1} \tag{12}$$

If the P_i matrix does not possess distinct eigenvectors, we can transform the matrix into the Jordan canonical form, which is composed of small Jordan blocks. For a typical Jordan block of order n which corresponds to a multiple eigenvalue r, we can find that e^{Pit} is also a Jordan block, with components of $e^{rt}, t e^{rt}, t^2/2!\, e^{rt}, t^{n-1}/(n-1)!\, e^{rt}$.

In this case, we can directly merge these n states into one and then apply our time-scale clustering method in the same way as the case of P that can be diagonalized.

In terms of the definition of eigenvalue and eigenvector, we have

$$(P_1 \otimes P_2)(Y_{1i} \otimes Y_{2j}) = P_1 Y_{1i} \otimes P_2 Y_{2j}$$

$$= r_{1i} Y_{1i} \otimes r_{2j} Y_{2j}$$

$$= (r_{1i}\, r_{2j})(Y_{1i} \otimes Y_{2j}). \qquad (13)$$

Therefore, the eigenvalues of $P_1 \otimes P_2$ are $r_{1i}\, r_{2j}$, with eigenvectors $Y_{1i} \otimes Y_{2j}$, $i = 1, 2, \ldots, n_1$ and $j = 1,2, \ldots, n_2$. The total number of eigenvalues is $n_1 n_2$.

Based on Eqn.'s (10) and (13), we can see that the n decaying time constants are $1 - r_{11}\, r_{21}$, $1 - r_{11}\, r_{22}$, ..., and $1 - r_{1n_1}\, r_{2\, n^2}$. The time constants of the aggregated traffic span a much larger range than the individual traffic. Note that for a practical size of traffic modeling by using MMPP, n = 20 ~ 30, there are many methods to find the eigenvalue easily. Also, there is no need to find the corresponding eigenvectors.

Now let's rearrange these time constants from the smallest to the largest and simply denote these constants by $r_1, r_2, \ldots, r_n$ that is, $r_1 \leq r_2, \leq \ldots \leq r_n$. The corresponding eigenvectors are denoted by $Y_1, Y_2 \ldots, Y_n$. The transformation matrix is thus

$$\Gamma = (Y_1, Y_2 \ldots, Y_n) \tag{14}$$

In order to aggregate these time constants into different groups of time scales, a natural choice is to use a log-scale, which also emphasizes the small time constants. Mathematically, among a group of time constants distributed over the log-scale, the one that has the shortest average (Euclidean) distance approximately represents the decaying property of the group. Thus, it is called RTC. The others can be represented by the RTC are called merged time constants (MTC's).

2.2 Traffic Aggregation By RTC's

In this section, we propose a new clustering technique that finds the most representative time constants. First, we need to determine how many RTC's are needed in order to approximate the original full-state MMPP process that describes the aggregated traffic process with satisfactory accuracy. Second, we need to determine which of the time constants are representative.

Assume that the time constants of the full-state MMPP process can be described by a random variable *x,* which is the Euclidean distance to the origin on the log-scale axis. In the case that *x* is non-uniformly distributed over the axis, i. e., *x* takes values around a few clustered points on the axis, the number of RTC's needed will be less than the one for the case when *x* is uniformly distributed. Therefore, the worst case would be that all the time constants are uniformly distributed over the axis.

In order to determine how many RTC's are needed in the worst case, we assume that *x* is uniformly distributed over the range of the time constants, which is $\Delta r = r_n - r_1$. We denote by *p* the probability that a time constant will be chosen as a RTC. The average number of RTC's is denoted by N_{RTC}. The average number of MTC's that will be merged into a RTC is denoted by N_{MTC}. With these notations, we can see that the *n* time constants will be grouped into *np* clusters, each cluster contains $n = (np)$ time constants in average, and one of them will serve as the RTC for this cluster. Therefore, we have

$$N_{RTC} = np$$

$$N_{MTC} = (n / np) - 1 = (1/P) - 1, \qquad (15)$$

The first requirement in state space reduction is to have less number of RTC's. The second one is to have each MTC within a RTC is closer to its RTC.

To meet both requirements, we design a combined cost function that measures both cost of the number of RTC's and the distance of an MTC to its RTC. Thus the cost function, denoted by *C(p)*, can be chosen as

$$C(p) = (1-\alpha) N_{RTC} + \alpha \sum_{j=1}^{N_{RTC}} N_{MTC} E\{| x_{MTC} - x_{RTC} |^2\} \quad (16)$$

Where $0 < \alpha < 1$ is a weighting factor that trades off the two requirements. A smaller value of α means higher approximation accuracy is emphasized. A larger value of means less number of RTC's is emphasized.

Based on the uniform distribution of x, we find that $E\{| x_{RTC} |\} = \Delta r / 2$ and $E\{| x_{MTC} - x_{RTC} |\} = \Delta r / (4np)$. Therefore,

$$C(p) = (1 - \alpha)np + \alpha np (1/p - 1) \Delta r^2 / (16 n^2 p^2) \quad (17)$$

By taking $dC(p) = dp = 0$, we find

$$p^3 + c_1 p = c_0 , \tag{18}$$

Where,

$$c_0 = (1 - \alpha)\Delta r^2 / (16 \alpha n^2)$$

$$c_0 = 2\, c_1 \tag{19}$$

Noting that the polynomial discriminant $D = (c_1/3)^3 + (c_0/2)^2 > 0$, one root is real and the other two are complex conjugates. In terms of the cubic formula, we have

$$p_{opt} = (c_1/3u) - u \tag{20}$$

Where,

$$u = \sqrt[3]{\left(- c_0/2 + \sqrt{(c_0^2 / 4 + c_1^3/27)}\right)}$$

It can be verified that $d^2C(p) = d^2p \,|_{p=popt} > 0$, which means $C(p)$ is indeed minimized at p_{opt}, which is the optimal probability for a time constant to be chosen as a RTC. Thus, we have the optimal number of RTC's:

$$N_{RTC} = np_{opt} \tag{21}$$

For the aggregation of two traffic streams with the same size of state-space $n_1 = n_2 = N$, based on Eqn.'s (20) and (21), we can even find a α value that satisfies $N_{RTC} \leq N$, i. e., the aggregation process does not increase the size of the state space. Thus, the state-space explosion problem can be solved if we aggregate the state-space according to the corresponding time scales.

After we obtain the N_{RTC} value, we can find out which of those time constants becomes RTC's by computer simulation. In this paper, we also develop a simple binary search algorithm to find RTC's.

Denote $\Delta x_i = x_i - x_{i-1}$, i = 1,2, …, n , as the increment between two consecutive time constants on the log-scale, i.e., $\Delta x_i = \log r_i - \log r_{i-1}$. Let's order the series from largest to smallest, we get x_1, x_2, ..., and x_n Denote the step counter as k and the number of RTC's found as n_k. Initially, set $k = 1$ and $n_1 = 2$ or $n_2 = 1$, which corresponds to the case of x_1 stands more closely to the rest of the points than to the origin. At each step, we chose two RTC's from the two clusters separated by a distance Δx_k. The point that stands closely to the center of a cluster will be selected as the RTC for the cluster. Within a cluster, we find the two sub-clusters that separated by the maximum distance among them. Again, in each of the sub-clusters, we chose the point that stands closely to the

center of the sub-cluster. Then, set $k = k + 1$ and $n_k = n_k + 2$. The searching continues until we find the total number of RTC's meets $n_{k+1} > N_{RTC}$ and $n_k < N_{RTC}$, where N_{RTC} is defined in Eqn. (21).

It is worth noting that the above algorithm also works for the case of non-uniformly distributed x. If the time constants are naturally clustered around some RTC's, we may find a good approximate model for the aggregated traffic. Otherwise, any approximation may work but not work well, if the approximation uses a reduced state space. As we mentioned before, in order to ensure that the reduced state space can approximate the original system model, the variance function has to be approximated with satisfactory accuracy, in addition to the marginal distribution. Therefore, we define a small constant $\in_0$ that measures the difference of the variance function caused by the reduced state space:

$$| V(N_t) - V_\alpha(N_t) | \leq \in_0 \qquad (22)$$

Where $V(N_t)$ is the variance of the reduced state space model, which can be evaluated by Eqn. (9). if the difference between the variance functions is small than $\in_0$, the reduced state space model results in the good approximation when it is used to predict system performance. Otherwise, we need to change the α value and find a different reduced state space model.

3. SIMULATION RESULTS

The simulations for this approximation algorithm is twofold (i) Matlab was used to test the effectiveness of the approximation algorithm by comparing the Queuelength probabilities (ii) NS-2 was used for the traffic generation and for evaluating the performance of the proposed approximation algorithm.

3.1 Queueing Analysis

Here we will now compare the queuelength probabilities for a matrix with 121 states and the reduced matrices of order 60 And 25 states. The superposition of two 11 states MMPP matrices yields another matrix, which is also of the type MMPP with 121 states. The queue length probabilities are computed for different values of buffer size *n* ranging from 0 to 100. Here while describing Fig (1) and Fig (2) we use the terms buffer size and queue length inter-changeably. Queue length probability is the probability that the packets are waiting in the buffer to receive service. In Fig (1) we show the comparison of the queue length probability with respect to the queue length or the size of the buffer. We can observe that the probability of packets waiting in the queue for traffic stream with 60 states is less than that of 121, but if we observe closely we find that this exists only when the size of the buffer is less than 30. Beyond a buffer size of 30 we can see that the queue length probability remains virtually the same irrespective of the

increase in the buffer size. This can be attributed to the reason that initially when the buffer size is less, the number of packets which arrive at the queue is a sizeable number compared to the buffer size, thus resulting in a lager value of queue length probability.

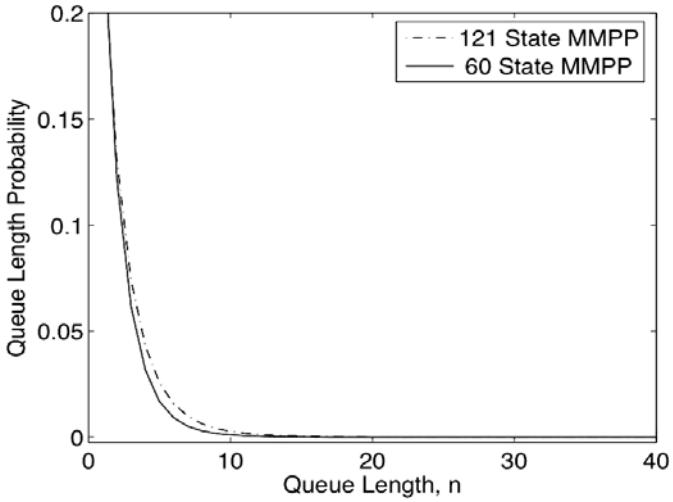

Fig (1) – Queue Length comparison for 121 and 60 state MMPP process

As the size of the buffer is being increased we observe a progressive decrease in the queue length probability, this is because the number of packets that are arriving at the queue are minute when compared with the size of the queue. Similarly in Fig (2) we can see that the original matrix with 121 states is approximated well by 25 states. Now again as when the buffer size is less the number of packets which are arriving at the queue is a sizeable number when compared with the length of the queue. As the queue length is being increased the number of packets is less compared to the length of the queue and so as the queue length is increased this ratio falls down and both the curves follow the same path.

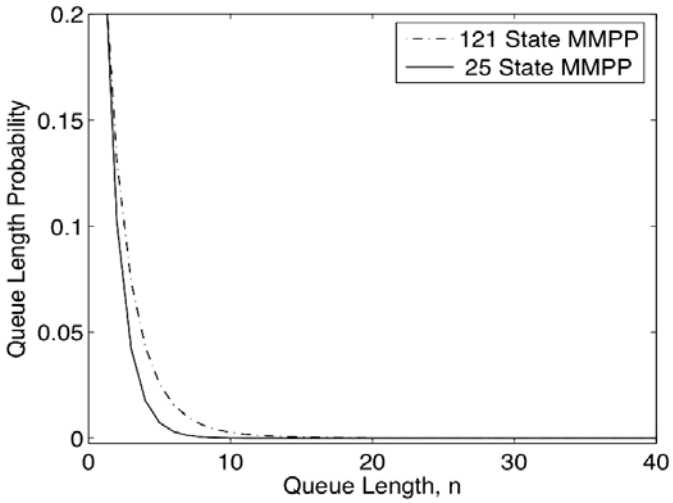

Fig (2) – Queue Length comparison for 121 and 25 state MMPP Process

3.2 An MMPP Traffic Model

The model used for this simulation was derived keeping in mind the human actions on a terminal interface that cause a sequence of events and behaviors of protocols at various layers of the protocol stack. For example, the generation of a request at the application level is translated into many transport level connections, and in turn each of these connections generates a sequence of data segments. These data segments are transported through the network layer through IP packets. Based on this we take into account three different time scales. A *flow* can be defined as a single TCP connection, initiated by a three-way handshake process. Using a closing procedure we can the TCP connection that has been established. Each *flow* generates a sequence of *packets*, which

are injected into the network. In fact, *flows* are not generated by themselves, rather by *sessions*. A *session* can be defined as the set of correlated *flows* that are injected into the network interface. Examples of session can be downloading less from the server connected through the means of FTP until it is disconnected, downloading some files from a web server in limited period of time; all the e-mail messages generated by a user that replays to all the previously downloaded e-mails or even a user connecting to the internet.

To generate a simple and generic model, which is, based only Markovian process, we adopt a poisson arrival process at the session level.

Sessions are generated according to Poisson process with arrival rate λ_s. Each session starts the arrival of a new flow. The number of flows generated by a session is a geometrically distributed random variable with mean equal to N_f. At the generation of the last flow the session ends.

Flows belonging to a session are generated according to poisson process with arrival rate λ_f. Each flow starts with generation of packets and the flow ends when N_p packets have been generated. N_p is the mean of number of packets to be generated which have a geometric distribution. *Packets* belonging to the same flow are generated according to poisson process with arrival rate λ_p.

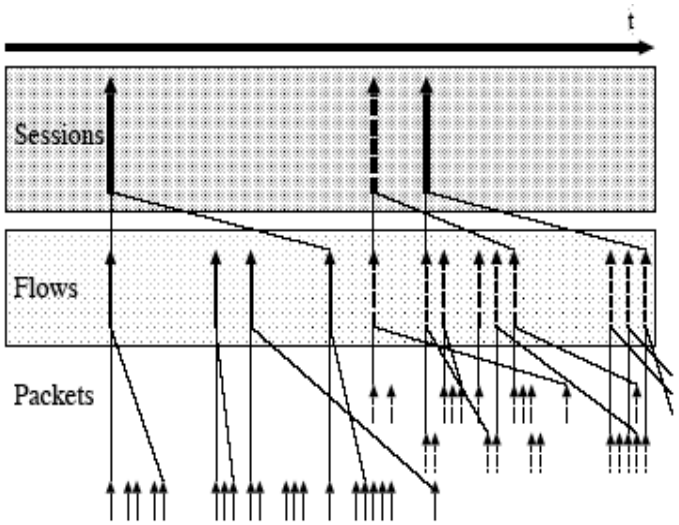

Fig (3) – Sessions, TCP Flows and Packets

Fig (3) shows the realization of the model we have been talking thus far. In this figure, three *sessions* arrive and each one in turn generates number of *flows* and each flow generated again generates number of *packets*, which will be multiplexed on the links along the source-destination paths.

Due to the above assumptions the packet and the flow arrival rates are also MMPP as discussed in [2]

1) Setting the Model Parameters: The model described above can be described by the following parameters:

- λ_s : arrival rate for new sessions;
- λ_f : arrival rate for new flows per active session;
- λ_p : arrival rate for packets per active flow;
- N_f : average number of flows generated per session;
- N_p : average number of packets generated per active flow;

As discussed in [2] three of the above parameters can be set based on the traces. To define the number of states for the traffic stream we vary the values of the number of active flows and the number of active sessions. The number of states of the traffic stream to be injected into the network can be decided using the following formula [(N_f x N_p) + number of sessions]. Using this formula we have varied the number of flows considering only 1 session. This means that all the flows are generated in one session. For example, say we want to generate a traffic stream of 121 states. For this we set N_f as 12, N_p as 10 and all the flows are generated in a single session. The transition from state (i, j) to (i − 1, j) implies the termination of a flow. But this does not mean that the flow terminated was the last flow of the session; its rate is $i\mu_f$, where $\mu_f = \lambda_p / (N_p - 1)$ and $\beta = 1 - 1/N_f$.

In a similar way, when the last flow of the session arrives, the session is not active anymore. Thus the there is a state transition from (i, j) to (i + 1, j − 1); this occurs at the

rate $j(1-\beta)\lambda_f$

In Fig (4) we show the plot of the traffic generated. The traffic has been generated for a 121 state MMPP process and the simulation time was 1000 seconds, if we observe the trace we can see that on an average the number of packets generated every time instant are between 150 and 200 packets. The traffic statistics have been shown in Table (1). It can be observed that the traffic described by an MMPP process is not very bursty as traffic described for IP links.

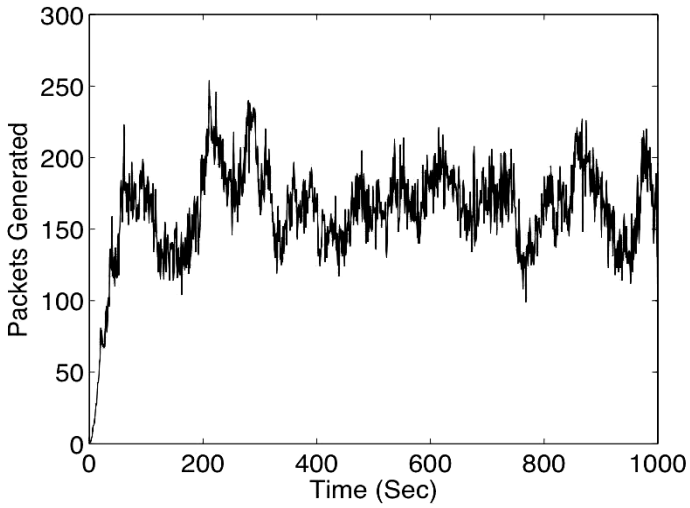

Fig (4) – MMPP Traffic Trace

On x-axis we have time, which is the time for which the packets have been generated. The total time for simulation was 1000s and on y-axis we have packets generated. We plot the packets generated at each instant of time.

Parameters	Collected Data
Number of Nodes used	2
Simulation Time	1000 seconds
Packet Size	500 bytes
Total Number of Packets generated	78732

Table (1) – Traffic Statistics

For analytical purposes, the traffic traces have been observed for two other cases as well. The traffic traces that have been generated for 200s and 500s can be seen in Fig(5) and Fig (6). From these traces it is evident that MMPP traffic appears burstier as we increase the length of simulation. This happens because as the length of simulations is increased more will be the packets that are generated from the flows, which are in turn generated by the sessions. When these traces are compared it is clear that the trace generated for 1000 seconds has maximum burstiness.

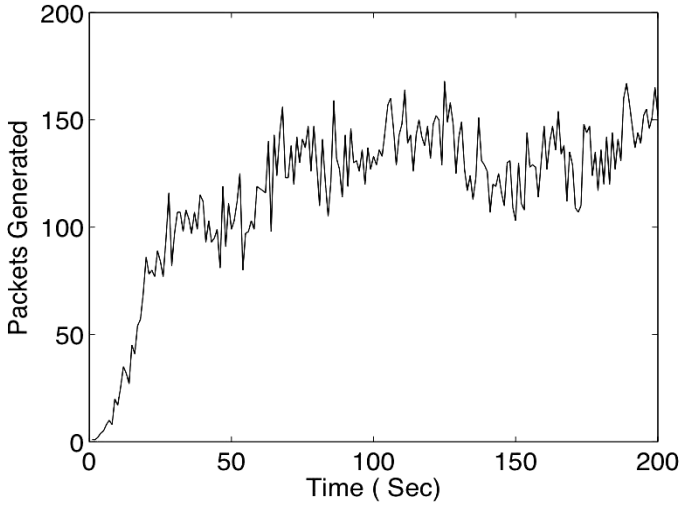

Fig (5) – MMPP Traffic Trace for 200

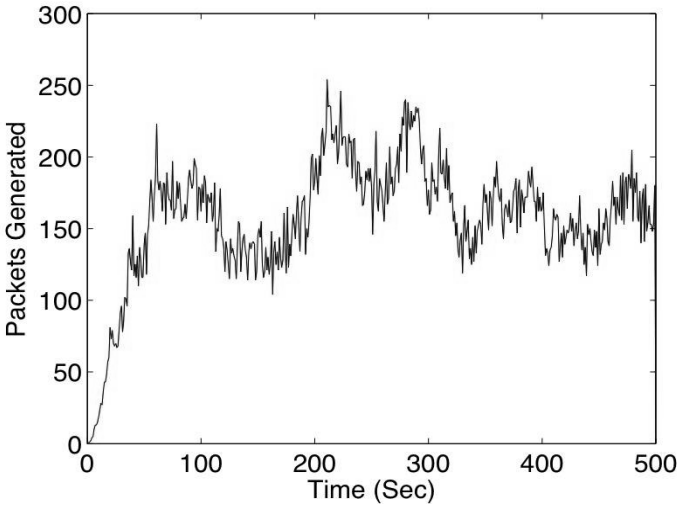

Fig (6) – MMPP Traffic Trace for 500

3.3 Performance Evaluation

The fidelity of the approximation process to MMPP with reduced states can be tested by estimating the mean packet delay. This test would reveal if the aggregated MMPP process is capturing the important characteristics of the original MMPP process. For the simulation of mean delay we use an infinite sized buffer which has constant service rate. We will now compare the mean delays for the original MMPP process and reduced states. As discussed in [1] Let the virtual waiting time and waiting time as seen by the arrival be W_V and W_A respectively. Then we have

$$E(W_V) = (3\rho - 2bQe) / (2(1 - \rho)) \quad (23)$$

$$E(W_A) = (1 – beQ) / \rho + \mathrm{E(V)} \quad (24)$$

Where $b = ((1-\rho)g + \pi Q)(e\pi + Q_0 + Q)^{-1}$, g is the stationary probability vector of the irreducible matrix G, which is a unique solution to the matrix functional equation

$$G = e^{(Q_0+QG)} \tag{25}$$

The matrix G is computed from (25), starting with the stochastic matrix $G_0 = e\pi$, for the later substitutions we use successive substitution method. In some cases it requires almost 40,000 iterations for the successive substitution process to converge. The convergence point for this iterative process has been set to ten decimal places. This computation takes no more than a couple of minutes on 2.8 GHz Dual Processor with 4 GB of RAM. The mean delay is then computed for different values of ρ, in Fig (6) we show the comparison of mean delay with respect to the traffic intensity, ρ. Using little's formula the mean queue lengths at an arbitrary time and at arrivals is given by $E(Q_V) = E(W_V)$ and $E(Q_V) = E(W_A)$, respectively.

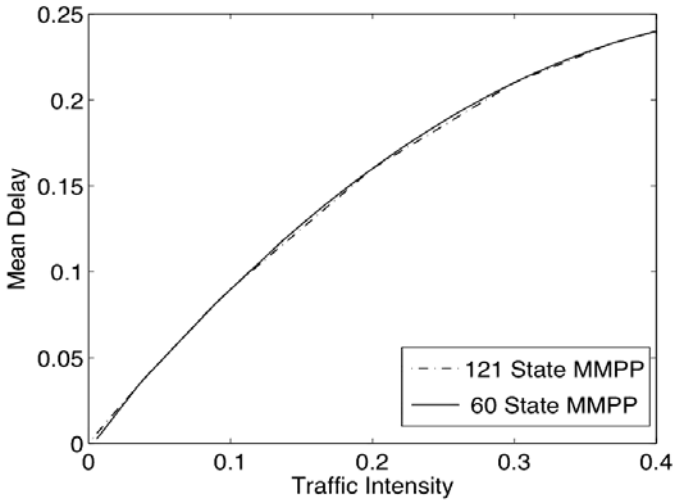

Fig (7) – Delay comparison for 121 and 60 state MMPP process

The second test to fidelity is the comparison of the packet loss ratio. In this case we take a finite buffer queue which is kept constant at n = 100, and simulate the packet loss ratio for different traffic intensities. Here we calculate the packet loss by computing the ratio of the number of packets dropped to the number of packets generated. It is known fact that when packets arrive at a rate faster than the rate at which they receive service from the queue, the queue will be filled with not many packets receiving service. This will eventually result in dropping of the packets from the queue. This is because the queue will be swamped with incoming packets and when the buffer is filled the packets that arriving later will be dropped. This tail drop will continue until the existing packets in the queue are being serviced and queue starts accepting packets again.

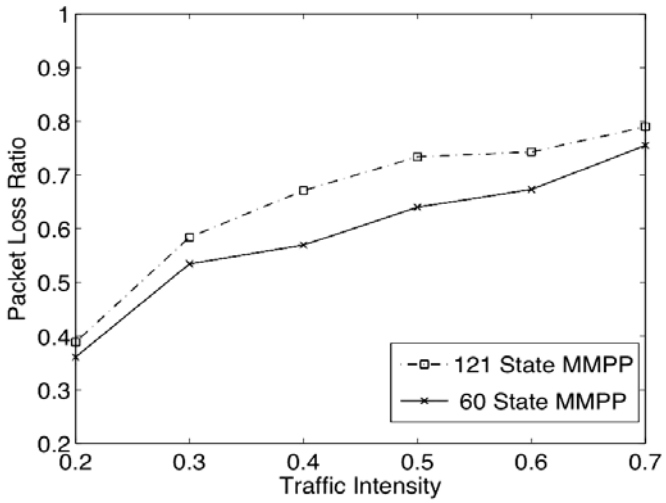

Fig (8) – Packet loss comparison of 121 and 60 state MMPP process

For our simulation we achieve tail drop by varying the traffic intensity ρ values from 0.2 to 0.7 . For this simulation we have collected traffic traces of the first 200s from the total runtime and the dropped packets have been observed from the generated traffic traces.

In the Fig (8) we show the packet loss ratio comparison for MMPP process's of states 121 and 60, we can observe that for the reduced process the packet loss is less when compared to the original MMPP process of 121 states. This shows that after approximation of 121 state MMPP process to a 60 state MMPP the number of packets that are being dropped are less when compared to actual process. Initially when the traffic intensity is at 0.2 and lesser, we see that the packet loss ratio is also very less. This is because when the traffic load is less we do not have many packets to be serviced. The packet loss ratio then steadily increases as the traffic intensity is increased to 0.3. The packet loss curves for the reduced model follows the shape of the original model, we can

see this clearly when the traffic intensity is increased from value of 0.3 to 0.6. Beyond this range of 0.6 as the traffic intensity is increased the packet drops increases. If the traffic intensity is further beyond this point it will fast approach the curve of 121 state model.

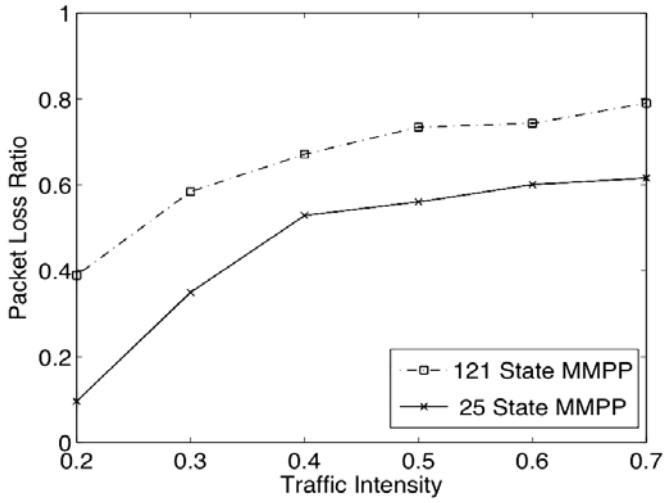

Fig (9) – Packet Loss ratio comparison for 121 and 25 state MMPP

In Fig (9) we show the comparison of the packet loss ratio between the 121 state MMPP and 25 state MMPP process. For this simulation again we have observed the traffic traces generated when the simulation was run. The traffic intensity was varied from 0.2 to 0.7. Similar to Fig (8), in Fig (9) also the packet loss ratio is lesser than the original MMPP model. When compared to the packet loss for 60 states, the 25 states model had a maximum of 0.5 packet loss ratio. Where as the 60 state model on an average was more than 0.6. This clearly shows an improvement of the performance for the reduced MMPP of 25 states. Here the curve for the reduced model almost maintains a path parallel to the 121 state MMPP process. Further reduction in the number of states might result in lesser packet loss ratio, but it could also result in error approximations.

This shows that the state space could be successfully reduced to almost 25 percent of the original number of states.

4. CONCLUSION

In this thesis, we find that if the time constants of the transition matrix are naturally clustered around some RTC's, we may find a good approximate model for the aggregated traffic. Our simulations have shown that the aggregated traffic provides a good match to the actual traffic stream, this has been shown in the queuelength comparisons between the aggregated and actual traffic stream. The performance evaluations have revealed that for the aggregated traffic the delay and packet loss are lesser than that for the actual stream indicating the improvement in performance when the traffic stream was aggregated. Also, the clustering of the time constants play an important role in the aggregation of the traffic stream. Thus the aggregation algorithm was effective in reducing the State-Space explosion.

5. REFERENCES

[1] D. P. Heyman and D. Lucantoni, "Modeling Multiple IP Traffic Streams With Rate Limits," *IEEE/ACM Trans. Networking*, vol. 11, no. 6, pp. 948-958, 2003.

[2] L. Muscariello, M. Mellia, M. Meo, R. Lo Cigno, M Ajmone Marsan, "A Simple Markovian Approach to Model Internet Traffic at Edge Routers," *Computer Communications 28 (2005).*

[3] A. T. Andersen and B. F. Nielsen, "A Markovian Approach for Modeling Packet Traffic with Long-Range Dependence," *IEEE Journal On Selected Areas in Communications*, vol. 16, no. 5, pp. 719-732, June 1998.

[4] A. Horvath and M. Telek, "A Markovian Point Process Exhibiting Multifractal Behavior and Its Application To Traffic Modeling," *Proc. IEEE INFOCOM*, 2003.

[5] M. Grossglauser and J. C. Bolot, "On the Relevance of Long-Range Dependence in Network Traffic," *IEEE/ACM Trans. Networking*, vol. 7, no. 5, pp. 629-640, 1998.

[6] P. Salvador, R. Valadas, and A. Pacheco, "Multiscale Fitting Procedure Using Markov Modulated Poisson Processes," *Telecommunication Systems*, 2003.

[7] W. Fischer and K. Meier-Hellstern, "The Markov-modulated Poisson Process (MMPP) Cookbook," *Performance Evaluation*, vol. 18, pp. 149- 171, 1992.

[8] H. Heffes and D. M. Lucantoni, “A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance,” *IEEE J. Select. Areas Commun.*, vol. SAC-4, no. 6, pp. 856-867, Sept. 1986.

[9] M. Yu, “A Traffic Aggregation and Switching Decomposition Technique for Network Performance Analysis,” *Technical Report*, AT&T Labs, May 2003.

[10] A. M. Viterbi, “Approximate analysis of time-synchronous packet networks,” *IEEE J. Select. Areas Commun.*, vol. SAC-4, no. 6, pp. 879-890, Sept. 1986 .

[11] I. Stavrakakis, “Efficient modeling of merging and splitting processes in larger networking structures,” *IEEE J. Select. Areas Commun.*, vol. SAC-9, no. 8, pp. 1336-1347, Oct. 1991.

[12] H. S. Lee, A. Bouhchouch, Y. Dallery, and Y. Frein, “Performance evaluation of open queueing networks with arbitrary configuration and finite buffers,” *Annals of Operations Research*, vol. 79, pp. 181-206, 1998.

[13] http://www.isi.edu/nsnam/ns/

Appendix

The Parameters of the 121-State Traffic Model

A1. The P Matrix:

Format: row 1, row 2, …, row 121.

The P matrix:

```
0.25 0.25 0000000000.25 0.25 0000000000000000000000000000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 00000000000000

0.05 0.225 0.225 000000000.05 0.225 0.225 000000000000000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000

00.1 0.2 0.2 000000000.1 0.2 0.2 0000000000000000000000000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 00000000000000

000.15 0.175 0.175 000000000.15 0.175 0.175 000000000000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000

0000.2 0.15 0.15 000000000.2 0.15 0.15 0000000000000000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 000000000000000000

00000.25 0.125 0.125 000000000.25 0.125 0.125 00000000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000

000000.3 0.1 0.1 000000000.3 0.1 0.1 0000000000000000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 00000000000000

0000000.35 0.075 0.075 000000000.35 0.075 0.075 00000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000

00000000.4 0.05 0.05 000000000.4 0.05 0.05 000000000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 0000000000000000

000000000.45 0.025 0.025 000000000.45 0.025 0.025 0000000000000000
 0000000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000
```

0000000000.5 00000000000.5 0000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
000000

0.05 0.05 0000000000.225 0.225 0000000000.225 0.225 000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000

0.01 0.045 0.045 000000000.045 0.2025 0.2025 000000000.045 0.2025
0.2025 0000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000

00.02 0.04 0.04 000000000.09 0.18 0.18 000000000.09 0.18 0.18 0000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000

000.03 0.035 0.035 000000000.135 0.1575 0.1575 000000000.135 0.1575
0.1575 0000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000

0000.04 0.03 0.03 000000000.18 0.135 0.135 000000000.18 0.135 0.135 0
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000

00000.05 0.025 0.025 000000000.225 0.1125 0.1125 000000000.225
0.1125 0.1125 0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000

000000.06 0.02 0.02 000000000.27 0.09 0.09 000000000.27 0.09 0.09 000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000

0000000.07 0.015 0.015 000000000.315 0.0675 0.0675 000000000.315
0.0675 0.0675 0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000

00000000.08 0.01 0.01 000000000.36 0.045 0.045 000000000.36 0.045
0.045 0000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000

000000000.09 0.005 0.005 000000000.405 0.0225 0.0225 000000000.405
0.0225 0.0225 0000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000

0000000000.1 00000000000.45 00000000000.45 0000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
0000000000

000000000000.1 0.1 0000000000.2 0.2 0000000000.2 0.2 0000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000000000

000000000000.02 0.09 0.09 000000000.04 0.18 0.18 000000000.04 0.18
0.18 0000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000

```
00000000000000.04 0.08 0.08 000000000.08 0.16 0.16 000000000.08 0.16
 0.16 0000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000000000000

000000000000000.06 0.07 0.07 000000000.12 0.14 0.14 000000000.12
 0.14 0.14 0000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000000000000000

000000000000000.08 0.06 0.06 000000000.16 0.12 0.12 000000000.16
 0.12 0.12 0000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000000000000000

0000000000000000.1 0.05 0.05 000000000.2 0.1 0.1 000000000.2 0.1 0.1
 0000000000000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000

00000000000000000.12 0.04 0.04 000000000.24 0.08 0.08 000000000.24
 0.08 0.08 0000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000000000000

00000000000000000.14 0.03 0.03 000000000.28 0.06 0.06 00000000
 0.28 0.06 0.06 0000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000000000000000

00000000000000000.16 0.02 0.02 000000000.32 0.04 0.04 00000000
 0.32 0.04 0.04 0000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000000000000000

00000000000000000.18 0.01 0.01 000000000.36 0.02 0.02 00000000
 0.36 0.02 0.02 0000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000000000000000

00000000000000000.2 00000000000.4 00000000000.4 000000000000
 0000000000000000000000000000000000000000000000000000000000000000000000000
 00000000

00000000000000000.15 0.15 0000000000.175 0.175 000000000
 0.175 0.175 0000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000000000000

00000000000000000.03 0.135 0.135 000000000.035 0.1575
 0.1575 000000000.035 0.1575 0.1575 0000000000000000000000000000000000
 00000000000000000000000000000000000000000000000000000000000000000000000000

00000000000000000.06 0.12 0.12 000000000.07 0.14 0.14 00000
 0000.07 0.14 0.14 0000000000000000000000000000000000000000000000000000000000000000
 00000000000000000000000000000000

00000000000000000.09 0.105 0.105 000000000.105 0.1225
 0.1225 000000000.105 0.1225 0.1225 0000000000000000000000000000000000
 00000000000000000000000000000000000000000000000000000000000000000000000000

00000000000000000.12 0.09 0.09 000000000.14 0.105 0.105 0
 00000000.14 0.105 0.105 0000000000000000000000000000000000000000000000000000
 00000000000000000000000000000000
```

```
000000000000000000000000000.15 0.075 0.075 000000000.175 0.0875
 0.0875 000000000.175 0.0875 0.0875 0000000000000000000000000000
 0000000000000000000000000000000000000000000

000000000000000000000000000.18 0.06 0.06 000000000.21 0.07 0.07 0
 00000000.21 0.07 0.07 00000000000000000000000000000000000000000
 000000000000000000000000000

000000000000000000000000000.21 0.045 0.045 000000000.245
 0.0525 0.0525 000000000.245 0.0525 0.0525 0000000000000000000000
 0000000000000000000000000000000000000000000

000000000000000000000000000.24 0.03 0.03 000000000.28 0.035
 0.035 000000000.28 0.035 0.035 0000000000000000000000000000000000
 000000000000000000000000000000000

000000000000000000000000000.27 0.015 0.015 000000000.315
 0.0175 0.0175 000000000.315 0.0175 0.0175 0000000000000000000000
 0000000000000000000000000000000000000000000

000000000000000000000000000.3 000000000000.35 0000000000
 0.35 00000000000000000000000000000000000000000000000000000000000
 000000000000

000000000000000000000000000.2 0.2 00000000000.15 0.15 0000
 000000.15 0.15 0000000000000000000000000000000000000000000000000
 0000000000000000

000000000000000000000000000.04 0.18 0.18 000000000.03
 0.135 0.135 000000000.03 0.135 0.135 0000000000000000000000000000
 000000000000000000000000000000000

000000000000000000000000000.08 0.16 0.16 000000000.06
 0.12 0.12 000000000.06 0.12 0.12 0000000000000000000000000000000
 000000000000000000000000000000

000000000000000000000000000.12 0.14 0.14 000000000.09
 0.105 0.105 000000000.09 0.105 0.105 0000000000000000000000000000
 000000000000000000000000000000

000000000000000000000000000.16 0.12 0.12 000000000.12
 0.09 0.09 000000000.12 0.09 0.09 0000000000000000000000000000000
 00000000000000000000000000

000000000000000000000000000.2 0.1 0.1 000000000.15
 0.075 0.075 000000000.15 0.075 0.075 0000000000000000000000000000
 00000000000000000000000000

000000000000000000000000000.24 0.08 0.08 00000000
 0.18 0.06 0.06 000000000.18 0.06 0.06 0000000000000000000000000
 000000000000000000000000000

000000000000000000000000000.28 0.06 0.06 00000000
 0.21 0.045 0.045 000000000.21 0.045 0.045 0000000000000000000000
 000000000000000000000000000
```

```
0000000000000000000000000000000000000000.32 0.04 0.04 00000000
 0.24 0.03 0.03 000000000.24 0.03 0.03 00000000000000000000000000000
 00000000000000000000000000000

0000000000000000000000000000000000000000000.36 0.02 0.02 0000000
 00.27 0.015 0.015 000000000.27 0.015 0.015 00000000000000000000
 00000000000000000000000000000000

0000000000000000000000000000000000000000000.4 00000000000.3 00
 000000000.3 00000000000000000000000000000000000000000000000000
 00000000

0000000000000000000000000000000000000000000.25 0.25 00000000
 00.125 0.125 0000000000.125 0.125 00000000000000000000000000000
 00000000000000000000

0000000000000000000000000000000000000000000000.05 0.225 0.225 00
 0000000.025 0.1125 0.1125 000000000.025 0.1125 0.1125 000000000000
 00000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000.1 0.2 0.2 000000
 000.05 0.1 0.1 000000000.05 0.1 0.1 00000000000000000000000000000
 00000000000000000000

0000000000000000000000000000000000000000000000.15 0.175 0.175
 000000000.075 0.0875 0.0875 000000000.075 0.0875 0.0875 000000000
 00000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000.2 0.15 0.15 00
 0000000.1 0.075 0.075 000000000.1 0.075 0.075 00000000000000000000
 00000000000000000000000000000

0000000000000000000000000000000000000000000000.25 0.125
 0.125 000000000.125 0.0625 0.0625 000000000.125 0.0625 0.0625 0000
 00000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000.3 0.1 0.1 00
 0000000.15 0.05 0.05 000000000.15 0.05 0.05 00000000000000000000
 00000000000000000000000000

0000000000000000000000000000000000000000000000.35 0.075
 0.075 000000000.175 0.0375 0.0375 000000000.175 0.0375 0.0375 0000
 00000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000.4 0.05
 0.05 000000000.2 0.025 0.025 000000000.2 0.025 0.025 0000000000000
 00000000000000000000000000000

0000000000000000000000000000000000000000000000.45
 0.025 0.025 000000000.225 0.0125 0.0125 000000000.225 0.0125
 0.0125 00000000000000000000000000000000000000000000

0000000000000000000000000000000000000000000000.5 0000
 0000000.25 00000000000.25 00000000000000000000000000000000000
 0000000000
```

0000000000000000000000000000000000000000000000000000000.3
0.3 0000000000.1 0.1 0000000000.1 0.1 0000000000000000000000000000
0000000000000000

0000000000000000000000000000000000000000000000000000000.06
0.27 0.27 000000000.02 0.09 0.09 000000000.02 0.09 0.09 00000000000
0000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000.12
0.24 0.24 000000000.04 0.08 0.08 000000000.04 0.08 0.08 00000000000
0000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
0.18 0.21 0.21 000000000.06 0.07 0.07 000000000.06 0.07 0.07 0000000
0000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
0.24 0.18 0.18 000000000.08 0.06 0.06 000000000.08 0.06 0.06 0000000
0000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
0.3 0.15 0.15 000000000.1 0.05 0.05 000000000.1 0.05 0.05 0000000000
0000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
0.36 0.12 0.12 000000000.12 0.04 0.04 000000000.12 0.04 0.04 0000000
0000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
00.42 0.09 0.09 000000000.14 0.03 0.03 000000000.14 0.03 0.03 000000
0000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
000.48 0.06 0.06 000000000.16 0.02 0.02 000000000.16 0.02 0.02 00000
0000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
0000.54 0.03 0.03 000000000.18 0.01 0.01 000000000.18 0.01 0.01 0000
0000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
00000.6 000000000000.2 000000000000.2 0000000000000000000000000000
00000000

0000000000000000000000000000000000000000000000000000000000
0000000.35 0.35 00000000000.075 0.075 00000000000.075 0.075 0000000
0000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
0000000.07 0.315 0.315 000000000.015 0.0675 0.0675 000000000.015
0.0675 0.0675 0000000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000
00000000.14 0.28 0.28 000000000.03 0.06 0.06 000000000.03 0.06 0.06
0000000000000000000000000000000

```
0000000000000000000000000000000000000000000000000000000000000000
 000000000.21 0.245 0.245 0000000000.045 0.0525 0.0525 00000000
 0.045 0.0525 0.0525 000000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000.28 0.21 0.21 000000000.06 0.045 0.045 000000000.06
 0.045 0.045 00000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 00000000000.35 0.175 0.175 000000000.075 0.0375 0.0375 00000000
 0.075 0.0375 0.0375 0000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000.42 0.14 0.14 0000000000.09 0.03 0.03 0000000000.09 0.03
 0.03 000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 00000000000000.49 0.105 0.105 0000000000.105 0.0225 0.0225 00000000
 0.105 0.0225 0.0225 0000000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 000000000000000.56 0.07 0.07 000000000.12 0.015 0.015 000000000.12
 0.015 0.015 00000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000.63 0.035 0.035 0000000000.135 0.0075 0.0075 000000
 000.135 0.0075 0.0075 000000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000.7 000000000000.15 000000000000.15 0000000000000
 0000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000.4 0.4 00000000000.05 0.05 00000000000.05 0.05 00
 0000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000.08 0.36 0.36 0000000000.01 0.045 0.045 00000000
 0.01 0.045 0.045 0000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000.16 0.32 0.32 0000000000.02 0.04 0.04 00000000
 0.02 0.04 0.04 0000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000.24 0.28 0.28 0000000000.03 0.035 0.035 000000
 000.03 0.035 0.035 000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000.32 0.24 0.24 0000000000.04 0.03 0.03 0000000
 00.04 0.03 0.03 000000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.4 0.2 0.2 0000000000.05 0.025 0.025 0000000
 00.05 0.025 0.025 000000000000000
```

```
0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.48 0.16 0.16 0000000000.06 0.02 0.02 00000
 0000.06 0.02 0.02 000000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.56 0.12 0.12 0000000000.07 0.015 0.015 00
 0000000.07 0.015 0.015 00000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.64 0.08 0.08 0000000000.08 0.01 0.01 000
 000000.08 0.01 0.01 000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.72 0.04 0.04 0000000000.09 0.005 0.005
 000000000.09 0.005 0.005 000000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.8 000000000000.1 000000000000.1 0000
 00000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.45 0.45 00000000000.025 0.025 0000
 000000.025 0.025 000000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.09 0.405 0.405 0000000000.005
 0.0225 0.0225 0000000000.005 0.0225 0.0225 00000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.18 0.36 0.36 0000000000.01 0.02
 0.02 0000000000.01 0.02 0.02 0000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.27 0.315 0.315 0000000000.015
 0.0175 0.0175 0000000000.015 0.0175 0.0175 000000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.36 0.27 0.27 0000000000.02
 0.015 0.015 0000000000.02 0.015 0.015 00000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.45 0.225 0.225 0000000000.025
 0.0125 0.0125 0000000000.025 0.0125 0.0125 0000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.54 0.18 0.18 0000000000.03
 0.01 0.01 0000000000.03 0.01 0.01 000

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.63 0.135 0.135 00000000
 0.035 0.0075 0.0075 0000000000.035 0.0075 0.0075 00

0000000000000000000000000000000000000000000000000000000000000000
 0000000000000000000000.72 0.09 0.09 0000000000.04
 0.005 0.005 0000000000.04 0.005 0.005 0
```

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000.81 0.045 0.045 00000000
0.045 0.0025 0.0025 0000000000.045 0.0025 0.0025

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000.9 000000000000.05 00000
000000.05 0

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.5 0.5 0000000000000000
000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.1 0.45 0.45 0000000000
0000000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.2 0.4 0.4 0000000000
00000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.3 0.35 0.35 0000000
0000000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.4 0.3 0.3 000000000
00000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.5 0.25 0.25 00000
0000000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.6 0.2 0.2 000000
00000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.7 0.15 0.15 000
0000000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.8 0.1 0.1 0000
00000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000.9 0.05 0.05 0
0000000000

0000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000010000000000
00

A2. The Rate Vector

diag(L)' =

0 6 12 18 24 30 36 42 48 54 60 6 12 18 24 30 36 42
48 54 60 66 12 18 24 30 36 42 48 54 60 66 72 18 24 30
36 42 48 54 60 66 72 78 24 30 36 42 48 54 60 66 72 78
84 30 36 42 48 54 60 66 72 78 84 90 36 42 48 54 60 66
72 78 84 90 96 42 48 54 60 66 72 78 84 90 96 102 48 54
60 66 72 78 84 90 96 102 108 54 60 66 72 78 84 90 96 102
108 114 60 66 72 78 84 90 96 102 108 114 120

A3: The Stationary Vector

pi =

0.00030072865982	0.00150364329911	0.00338319742299
0.00451092989733	0.00394706366016	0.00236823819610
0.00098676591504	0.00028193311858	0.00005286245973
0.00000587360664	0.00000029368033	0.00150364329911
0.00751821649554	0.01691598711497	0.02255464948663
0.01973531830080	0.01184119098048	0.00493382957520
0.00140966559291	0.00026431229867	0.00002936803319
0.00000146840166	0.00338319742299	0.01691598711497
0.03806097100869	0.05074796134491	0.04440446617680
0.02664267970608	0.01110111654420	0.00317174758406
0.00059470267201	0.00006607807467	0.00000330390373
0.00451092989733	0.02255464948663	0.05074796134491
0.06766394845989	0.05920595490240	0.03552357294144
0.01480148872560	0.00422899677874	0.00079293689601
0.00008810409956	0.00000440520498	0.00394706366016
0.01973531830080	0.04440446617680	0.05920595490240
0.05180521053960	0.03108312632376	0.01295130263490
0.00370037218140	0.00069381978401	0.00007709108711
0.00000385455436	0.00236823819610	0.01184119098048
0.02664267970608	0.03552357294144	0.03108312632376
0.01864987579426	0.00777078158094	0.00222022330884
0.00041629187041	0.00004625465227	0.00000231273261
0.00098676591504	0.00493382957520	0.01110111654420
0.01480148872560	0.01295130263490	0.00777078158094
0.00323782565873	0.00092509304535	0.00017345494600

0.00001927277178 0.00000096363859 0.00028193311858
0.00140966559291 0.00317174758406 0.00422899677874
0.00370037218140 0.00222022330884 0.00092509304535
0.00026431229867 0.00004955855600 0.00000550650622
0.00000027532531 0.00005286245973 0.00026431229867
0.00059470267201 0.00079293689601 0.00069381978401
0.00041629187041 0.00017345494600 0.00004955855600
0.00000929222925 0.00000103246992 0.00000005162350
0.00000587360664 0.00002936803319 0.00006607807467
0.00008810409956 0.00007709108711 0.00004625465227
0.00001927277178 0.00000550650622 0.00000103246992
0.00000011471888 0.00000000573594 0.00000029368033
0.00000146840166 0.00000330390373 0.00000440520498
0.00000385455436 0.00000231273261 0.00000096363859
0.00000027532531 0.00000005162350 0.00000000573594
0.00000000028680

A4: The Time Constants

0.0000	0.9000	1.0500
0.1500	0.9000	1.0500
0.1500	0.9150	1.0500
0.2775	0.9150	1.0500
0.3000	0.9300	1.0500
0.3000	0.9300	1.0800
0.4050	0.9300	1.0800
0.4050	0.9300	1.0875
0.4500	0.9375	1.0875
0.4500	0.9450	1.1100
0.5100	0.9450	1.1100
0.5325	0.9600	1.1250
0.5325	0.9600	1.1250
0.6000	0.9600	1.1400
0.6000	0.9750	1.1400
0.6150	0.9750	1.1400
0.6150	0.9750	1.1400
0.6600	0.9750	1.1700
0.6600	0.9825	1.1700
0.6975	0.9825	1.1925
0.7200	0.9900	1.1925
0.7200	0.9900	1.2000
0.7500	0.9900	1.2000
0.7500	0.9975	1.2000
0.7500	1.0050	1.2000
0.7800	1.0050	1.2450
0.7800	1.0125	1.2450
0.7875	1.0125	1.2750
0.7875	1.0200	1.2750
0.8250	1.0200	1.2975
0.8250	1.0200	1.2975
0.8250	1.0200	1.3500
0.8250	1.0275	1.3500
0.8400	1.0275	1.3500
0.8625	1.0350	1.3500
0.8625	1.0350	1.4250
0.8775	1.0350	1.4250
0.9000	1.0350	1.5000
0.9000	1.0425	1.5000
0.9000	1.0425	
0.9000	1.0500	