Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 340 (2018) 101–121

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Interactive Behaviour Between the Dynamic Arrival Process of Complex Traffic Source and a Bottleneck Queue

Zhenyu Chen [1]

Wolfson School of Mechanical, Electrical and Manufacturing Engineering
Loughborough University
Loughborough, LE11 3TU, UK

Lin Guan [2]

Department of Computer Science
Loughborough University
Loughborough, LE11 3TU, UK

Xingang Wang [3]

Department of Computing
Coventry University
Priory St, Coventry, CV1 5FB, UK

Abstract

This paper proposes a discrete-time heterogeneous model to represent the interaction between the dynamic arrival process of multiple TCP sources and a bottleneck queue. It utilises a novel time-variant calculation approach for transient queueing models, which modifies the conventional steady-state method by no longer requiring the equilibrium assumption. The main contributions are: (1) The limitation of the conventional queueing method is analysed and the benefits of the proposed modification is demonstrated; (2) The proposed dynamic arrival process of multiple sources is illustrated, which interacts with the queueing process and statistically responds to the positive and negative congestion feedback; (3) The fidelity of the proposed model is extensively validated by the closely matched results from MATLAB numerical solutions and NS-2 discrete-event simulation experiments.

Keywords: network performance, discrete-time queuing model, heterogeneous queue, transient states, TCP dynamic arrival process

[1] Email: z.chen5@lboro.ac.uk

[2] Email: l.guan@lboro.ac.uk

[3] Email: aa9038@coventry.ac.uk

https://doi.org/10.1016/j.entcs.2018.09.008

1571-0661/© 2018 The Author(s). Published by Elsevier B.V.

1 Introduction

It is common practice to use queueing theory to solve the problems in the performance modelling of communication and computer networks. There is much research that aims to extend the fundamental Geo/Geo/1 (discrete-time M/M/1) queueing model [24][3]. Most of the work focuses on replacing the basic arrival and departure model with a more complicated and more accurate one[31][9]. Almost all of the derived models still utilise a similar method for calculation, which assumes the queueing system is in a steady state - equilibrium, and solves the balance equations to acquire the steady state probability distribution for the queueing system[18][5]. However, the authors believe that requiring a steady state is a significant limitation, since a typical practical network system is unlikely to remain in this state, due to the burstiness nature of the Internet traffic [20]. As a result, a new modification to the conventional method is proposed which does not require the steady state assumption for the calculation of the state probability distribution.

There is other research of seemingly similar topics about the queue before equilibrium, such as [10][17][30]. Most of them, however, focus on continuous-time queueing systems, whereas this paper, a discrete-time model is selected, due to the simplification it brings to the representation of timing of event within a discrete time slot. Much of the current research also focuses on the mathematical properties of the transient behaviour of such queueing system, such as the convergence rate [15][4] or transient probabilities [11][16]. In comparison, this paper introduces a practical modification the the conventional method that allows the performance of the queueing system be obtained before the equilibrium state is reached; this modification is then utilised in the development of interactive network model where its simplicity as well as time-variant nature is crucial.

While many complex mathematical models have been proposed to represent Internet traffic, most of them actually focus on the burstiness and correlation features of traffic source itself, such as the superposition of MMPP[21]. To the best of the authors' knowledge, few of them actually considered the dynamic interaction between the source and the queue, most prominently, the Additive Increase Multiplicative Decrease (AIMD) feature of TCP, which is a dominant feature in the practical network systems. There is also research that aims to develop precise model for TCP, most notably, the fluid model [26][19], which takes the dropping event into consideration. However, most of them fail to establish an elaborate queueing model of the bottleneck queue to analyse the probability of the dropping events, instead relying on some form of approximation or simplification that rarely represents that dynamic interaction. The proposed model in this paper also aims to address this issue.

In a previous research, an attempt to modify the conventional method was made in order to create a similar interactive model with only a single traffic source. It did not provide a generic representation, and the single-source assumption severely limits the usefulness of the model [7]. This research improves significantly on both of these aspects, allowing the modification to be used in different models with its generic form, while incorporate multiple traffic sources using a poisson binomial distribution which allow varying transmitting rate from different traffic sources.

The remainder of this paper is organised as follows: Section 2 analyses the limitation of using the equilibrium assumption in the conventional discrete-time queueing method while demonstrates the proposed modification which overcomes this; Section 3 introduces the new multiple-source dynamic arrival process, which interacts with the queue and responses to the network events; Section 4 provides evidence of effectiveness and accuracy for the proposed model by performing evaluation with closely matched results between numerical solutions using MATLAB and software simulation experiments using NS-2, under various parameters; finally, Section 5 concludes the current work and explain the plans for future work.

2 Time-variant Calculation Method

2.1 Limitation of conventional method

In the conventional method, an equilibrium assumption is required. It is usually arbitrarily assumed to be possible to reach and referenced several times through the calculation process, which usually involves solving the steady state probability distribution for the queue length process by deriving the transitional probabilities between different states. The normalising equation is then used to gather the probability for a particular state and in turns the distribution for all other states can be derived from that[29]. In order to reach this state, for an irreducible, aperiodic and positive recurrent discrete-time Geo/Geo/1 Markov chain, the arrival rate is required to be strictly less than the service rate[29].

The principle idea behind the conventional method trying to obtain the state probability distribution at the equilibrium state is that in many practical system, it is the eventual status of the system that people usually care about. Conventional equilibrium state method is great for this task when the parameters of the queueing system remain static long enough so that it can reach the equilibrium state (assuming that its parameters lead to convergence). However, when this is not the case, using the conventional equilibrium state method is no longer effective.

In the practical network system, especially when involving commercial Internet connections, the maximal potential arrival rate is very likely to be greater than the mean service rate of the bottleneck gateway over the connection [12], at least during certain phase. As it is extremely unlikely that many network users peak their network traffic at the exact same time, in order to keep the utilisation rate of the bandwidth resource at a acceptably high level, the bandwidth/maximal throughput of a long-distance connection is reasonably made to be smaller than the sum of those of all the edge links that it connects to [22]. As a TCP source identifies the congestion if it detects a loss event over this connection and responds by decreasing transmission speed, the overall mean traffic arrival rate for the entire connection will be much less than the peak transmission rate. If a number of greedy TCP sender is sharing a bottleneck gateway, it is extremely likely that the global arrival rate is greater than the departure rate during some portion of time, resulting in an increase of queue length; this would then be followed by the global arrival rate smaller than the departure rate during some other portion of time due to loss event being detected by the TCP senders [25]. With the conventional steady-state method, this behaviour has to be simplified as this has obviously not reached equilibrium.

2.2 Demonstration of the proposed method

The discrete-time queueing model does not limit to the balance state, but the conventional analysis and calculation method based on equilibrium balance equations does. The core form for the balance equations on which the conventional method is based on can be written as

$$\Pi_k = \sum_{i=0}^{J} \Pi_i p_{ik}, \tag{1}$$

where p_{ik} is the transition probability from state i to state k, J is the size of the finite queue capacity. This, is a special form of equation

$$\Pi_k(t+1) = \sum_{i=0}^{J} \Pi_i(t) p_{ik} \tag{2}$$

where variables $(t+1)$ and (t) represent the time slot index. This equation is also known as the Chapman-Kolmogorov equation[8]. When the system is in equilibrium, state probability distribution of adjacent time slot are the same, thus the time slot index $(t+1)$ or (t) is no longer relevant. Replacing both hand sides of Equation (2) with a common equilibrium state probability distribution Π_k, and it becomes the common form of balance equation of the conventional method. When the system is not in equilibrium state, this derived equation calculates the state probability distribution for the queue in the next time slot based on the distribution values of the current state.

Based on the type of the queue used in the modelling, certain state of the system can only be reached by transiting from a limited number of states from the previous time slot, with a certain probability that is positive and usually calculable. For example, in a Geo/Geo/1/J queue, as no more than one arrival event and no more than one departure event can happen in a single time slot, state 0 can only be reached by transiting from state 0 or 1 from the previous time slot, with the probability of no departure no arrival and one departure no arrival respectively. This is the situation in an Early Arrival System (EAS), observed from the centre of the time slot [8]. In this way, the summation operation in Equation (2) can be simplified as transitions with non-positive probability can be omitted. The new set of equations can be written as

$$\begin{aligned}\Pi_{0(t+1)} =&\Pi_{0(t)}(1-\alpha)+\Pi_{1(t)}(1-\alpha)\beta\\ \Pi_{1(t+1)} =&\Pi_{0(t)}\alpha+\Pi_{1(t)}[\alpha\beta\\ &+(1{-}\alpha)(1-\beta)]+\Pi_{2(t)}(1-\alpha)\beta\\ &\vdots\\ \Pi_{k(t+1)} =&\Pi_{k-1(t)}\alpha(1-\beta)+\Pi_{k(t)}[\alpha\beta\\ &+(1{-}\alpha)(1-\beta)]+\Pi_{k+1(t)}(1-\alpha)\beta,\\ &k=2,3,4,\cdots,J-1\end{aligned} \tag{3}$$

where $\Pi_{i(t)}$ represents the probability of the queue in state i (with a queue length of i) in time slot t, α represents the arrival rate, β represents the service rate.

Without the steady-state assumption, the time slot variation has to be considered, and it is not possible to solve this set of equations by simply using the normalising equation. However, this set of equations does provide a way to calculate the state probability distribution of the system in the next time slot, given the distribution of the current state is known. It also means that it is possible to calculate the distribution recursively as long as a time slot with a known state can be identified. Because it is always possible to start the calculation from the empty queue state (which is a known state as the system is 100% in state 0), it is possible to use this method to deduce the queue length distribution of the system within any future time slot.

If this method is used to calculate a typical homogeneous Geo/Geo/1/J queue, as the transitional probability for every state are identical over the time, the state probability distribution of the queue can be written in general as

$$\begin{aligned}P(t) =&P(t-1)p_{(t-1)}=P(t-2)p_{(t-2)}p_{(t-1)}\\ =&P(0)p_1p_2\cdots p_{(t-2)}p_{(t-1)}\end{aligned} \tag{4}$$

where $P(t)$ is the state probability distribution $\{\Pi_1,\Pi_2,\cdots,\Pi_J\}$ of the queue at time slot t, and p_t is the state transition matrix for the queue at time slot t. For a homogeneous Geo/Geo/1/J queue with arrival rate α and departure rate β, p_i always equals to

$$\begin{bmatrix} 1-\alpha & \alpha & 0 & \cdots & 0\\ \beta(1-\alpha) & \alpha\beta+(1-\alpha)(1-\beta) & \alpha(1-\beta) & \cdots & 0\\ 0 & \beta(1-\alpha) & \alpha\beta+(1-\alpha)(1-\beta) & \cdots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & \alpha\beta+(1-\beta)\end{bmatrix} \tag{5}$$

In this special case, the state probability distribution k time slots after the initial queue state where it is empty can be represented as

$$P(k) = P(0)p^k = p^k \tag{6}$$

where p is the state transition matrix for the queue, which is represented in Matrix (5). This is a special form for the n-step transition probabilities equation.

With Equation (4), there is no special assumption requirements, namely, the equilibrium limitation of the conventional method. It allows easier analysis for the transient states at the expense of requiring to calculate state probability distribution of all the intermediate time slots. More importantly, the state transition matrix for each time slot does not require to be the same value, allowing a time-variant model with different arrival and departure parameter, or even entirely different arrival and departure process model to be used at any time slot. If used within the same Geo/Geo/1/J model, it allows simply varying the arrival and/or departure rate parameter(s) α, β in each time slot, and the complexity increase is easily manageable. This property allows the development of a time-variant interactive model to represent the TCP/AQM source/queue dynamic interactivity in Section 3.

3 Dynamic Arrival Process Representing Multiple TCP Traffic Sources

This section presents a dynamic time-variant arrival process which behaves like multiple TCP traffic sources with different parameters, with the additional option of including unresponsive source, representing UDP traffic in a practical system. It would be utilised as the arrival process for the time-variant queueing systems. One of the most important reasons for the difficulty of representing the Internet traffic with source model without the time-variant approach is that these models are usually based on steady state analysis [20]. As it has been discussed before, in a practical network system with dynamic traffic flows, equilibrium state, or steady state, is often not an appropriate assumption. Using the time-variant calculation method proposed in Section 2, this is no longer the concern.

3.1 Modelling process for TCP dynamic transmission rate

This subsection demonstrates the method of representing TCP traffic as its transmission rate varies in response to the network positive and negative congestion feedback. TCP employs flow control and congestion avoidance features. It increases the sizes of flow control window and congestion window in order to transmit data packets as fast as possible and keep the bandwidth resource utilisation rate high, while avoiding congestion by decreasing the window sizes when loss events are detected [1]. If the modelled queue is the bottleneck of the TCP connection, then the transmission rate, which is directly affected by the two window sizes, has a direct relationship with the throughput performance of that queue and gateway [6].

The TCP source transmission rate changes over time, and thus can not be easily modelled by a model that is invariant over time.

In the Internet architecture, transport layer is responsible for the control of transmission speed. At a lower layer, the hardware system involves during data transmission has a generally fixed internal working clock and cycle, and on average it would take the same amount of time to process and transmit a packet of the same size, with some variation due to unforeseeable internal physical and external environmental reasons[23]. The packet generation interval at this layer is largely memoryless and could be very well modelled by a basic Bernoulli process. However, the involvement of transport layer protocol, namely TCP, makes the lower layers not always in the "active transmission stage". Instead, when the TCP source detects that the flow control or congestion window is full, the data transmission stop completely for this session. This "idle stage" can still be represented by a Bernoulli model, albeit with a data packet transmission rate of 0. Modelling both states with a same model can greatly simplify the process, as it would only be the parameter changing from different time slots.

In this case, the complicated TCP source dynamic behaviour can be represented by a simple Bernoulli process with varying parameters at different transmission stage, which is influenced by the feedback from the queue. Modelling in this way is an even better representation of the packet transmission process in the physical hardware and controlling software protocol comparing to many others where the model simply transits arbitrarily from one state to another with no or few weak justification.

The model represent the size of congestion and flow control windows (which directly affect the transmission rate) probabilistically, while representing the transmission rate in different states deterministically. The term deterministically here indicates that for each time slot, the source must be in either one of the transmission stage, active or idle, determined by the probabilistically calculated window size, and the transmission rate in each state can be directly derived from the network specification. From the perspective of the bottleneck queue, the arrival rate during the active transmission stage is associated with the link bandwidth between the source and the bottleneck gateway and is zero during the idle state, while the departure rate is associated with the link bandwidth between the bottleneck gateway and the destination, assuming that the hardware processing speed is always much faster than the link speed, which is usually a valid assumption in modern computer network system. It is a logic choice to make the link bandwidth between the source and the bottleneck gateway greater than the link bandwidth between the bottleneck gateway and the destination, which means the arrival rate in the active transmitting state to be greater than the departure rate, as it would be pointless to study queueing theory for congestion control if this is not the case as congestion will never occur [14]. In addition, it is also an accurate representation of the practical network system.

As a result, the transmission stage that a source is in is determined by its congestion window size and the stage in turn determines the overall observed arrival rate from the perspective of the bottleneck queue. This makes congestion window size modelling one of the most critical parts. In this research, the congestion window size of a TCP source is modelled after the specification of TCP-Reno [2].

All possible changes to window size happen over at least one round-trip time (RTT). In each RTT, there is a possibility of either no loss event of any kind is detected, or the opposite. In the former case, the window size increases by a specified size if an positive acknowledgement packet (ACK) is received and maximal window size s_{max} has not been reached yet (or remains if s_{max} has been reached); in the latter case, loss event is detected in forms of triple duplicate ACKs or timeout, and the window size decreases by a specified size. The congestion window size can be represented recursively as

$$\begin{aligned} S_{(T+1)} =& S_{(T)} + p_{dc}S_{DV} \\ &+ p_{ic}S_{IV} \end{aligned} \tag{7}$$

where p_{dc} and p_{ic} are the probabilities for congestion window size decrease and increase, respectively, S_{DV} and S_{IV} are the congestion window size decreasing and increasing value, should the corresponding condition is satisfied. $S_{(T+1)}$ and $S_{(T)}$ are the statistical congestion window sizes, where $\{T, T+1, \cdots\}$ are batches of time slots divided by RTT. These batches are of similar size if the RTT of the packets are not oscillating heavily and no loss event of any kind are detected. However, the loss event may be detected at any random point inside or on the edge of a time slots batch, which will end this batch immediately and change the congestion window size, and restart the RTT count. According to the TCP specification, in the congestion avoidance phase, congestion window size increases by 1 MSS for every RTT with no loss event of any form. In slow start phase, this becomes increasing by 1 MSS for every positive ACK received. This leads to the following conclusions:

$$S_{IV} = \begin{cases} S_{pkt} & \text{(congestion avoidance);} \\ S_{(T)} & \text{(slow start).} \end{cases} \tag{8}$$

$$p_{ic} = \alpha p_{\overline{d}} = \alpha[1 - (\Pi_J + p_a)] \tag{9}$$

where S_{IV} and p_{ic} are the congestion window size increase value and probability used in Equation (7), S_{pkt} is the data packet size, Π_J is the probability that the queue is in state J, in another word, full, and p_a is the probability for any active dropping queue management algorithm to drop that the packet.

As for the detected loss events, there is

$$S_{DV} = -\frac{S_{(T)}}{2} \tag{10}$$

$$p_{dc} = p_d = \Pi_J + p_a \tag{11}$$

Here the dropping probability p_d of the queue is used as the probability of window size decrease, while in practice triple duplicate ACK or timeout is the loss events. This is because the those two events can generally be consider equivalent.

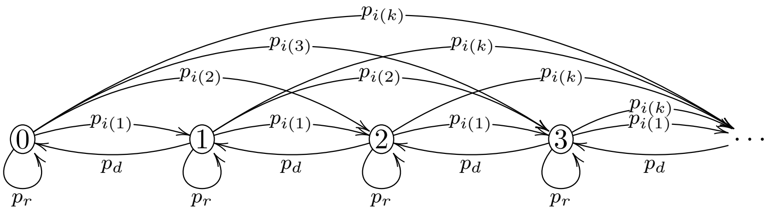
Combining Equations (7)(8)(9)(11)(10), the window size can be calculated, thus the transmission status (either the source is in the "active transmission stage" or the "idle stage") during each time slot can is determined by comparing the sum of the size of un-acknowledged packets and the current congestion window size, and thus the model for an entire TCP connection become a combination of many stages, each of that stage can be represented by a Geo/Geo/1/J queueing model with different parameters (most importantly, α), and the transitions between different stages are determined by the status of the congestion window size, which is in turn affected by the performance measure (namely, the packet dropping rate) of the queue, quite similar to the way TCP source behaves in practical systems. In this way, the model is able to capture most aspects of a dynamic TCP source behaviour while still be simple enough to for calculation and analysis with time-variant queueing method proposed in Section 2.2.

3.2 Interaction between multiple complex sources

In this section, large numbers of traffic sources with different configurations and even unresponsive traffic sources, representing the UDP traffic, and their interaction is represented in the model.

In practical network systems, it is most likely that the bottleneck routers on the core network have a much higher service rate comparing to the traffic transmission rate of individual network users, and only because of the very large number network users connecting through the same core routers and a large potion of the users actively transmitting packets at the similar time, can severe congestion be observed.

To precisely reflect this situation, it is logical to construct a system model which allows multiple arrival in a time slot. For modelling a single-service system, the maximal one departure model is still competent. The size of the time slot can be defined as similar to the service rate to keep the one departure model effective. This type of model is usually labelled as $M^{a_n}/M/1$ model [29], and it is sometimes used in conjunction with the conventional equilibrium state method. The queue length state transitional diagram for a discrete-time $M^{a_n}/M/1$ queue can be drawn as below.



(12)

where $p_{i(k)}, k = \{1, 2, 3, \cdots, n\}$ is the probability that the queue length increases k, p_r is the probability that queue length remains, and p_d is the probability that queue length decrease.

According to the assumptions, p_d and p_r could be represented easily. As there is only one possible departure in any time slot, while multiple possible arrival, in order for queue length to decrease, all possible arrival events must be absent while the departure event occurs. Assuming the probability that packet arrives from each source to be $\alpha_1, \alpha_2, \alpha_3, \cdots, \alpha_n$, and probability that packet depart from the queue in a time slot to be β, this means

$$\begin{aligned} p_d =& \beta(1-\alpha_1)(1-\alpha_2)(1-\alpha_3)\cdots(1-\alpha_n) \\ =& \beta\prod_{k=1}^{n}(1-\alpha_k) \end{aligned} \tag{13}$$

where n is the total number of traffic sources transmitting through this queue. In order for queue length to remains over the time length, there can be either one arrival and one departure, or no arrival and no departure. This leads to

$$\begin{aligned} p_r =& p_{0a0d} + p_{1a1d} \\ p_{0a0d} =& (1-\beta)\prod_{k=1}^{n}(1-\alpha_k) \\ p_{1a1d} =& \beta[\alpha_1(1-\alpha_2)(1-\alpha_3)\cdots(1-\alpha_n) \\ & + (1-\alpha_1)\alpha_2(1-\alpha_3)\cdots(1-\alpha_n) \\ & \quad \vdots \\ & + (1-\alpha_1)(1-\alpha_2)(1-\alpha_3)\cdots\alpha_n] \\ =& \beta\sum_{i=1}^{n}\frac{\alpha_i\prod_{k=1}^{n}(1-\alpha_k)}{1-\alpha_i} \end{aligned} \tag{14}$$

where p_{0a0d} represents the probability of no arrival and no departure in a time slot, and p_{1a1d} represents the probability of one arrival and one departure.

As for the probability that queue length increases to a certain higher valued state, say $+N$, all of them can be either no departure and N packets arrive, or one departure and $(N+1)$ packets arrive. If $\alpha_1 = \alpha_2 = \cdots = \alpha_n = \alpha$, this arrival can be described as a binomial distribution [27], and its probability can be easily deduced as

$$\begin{aligned} p_i(k) =& p_{ka0d} + p_{(k+1)a1d} \\ p_{ka0d} =& (1-\beta)\binom{n}{k}\alpha^k(1-\alpha^{n-k}) \\ p_{(k+1)a1d} =& \beta\binom{n}{k+1}\alpha^{k+1}(1-\alpha^{n-k-1}) \end{aligned} \tag{15}$$

In a more generic case, if not every arrival rate is the same, it becomes the so-called Poisson binomial distribution [28]. The probability mass function (pmf) of having k successful trials out of a total of n with the Poisson binomial distribution can be generally written as

$$P(K=k) = \sum_{A\in X_k}\prod_{i\in A}p_i\prod_{j\in A^c}(1-p_j) \tag{16}$$

where X_k is the set of all subsets of k integers that can be selected from $\{1, 2, 3, \cdots, n\}$, and A^c is the complement of A.

The calculation for the Poisson Binomial distribution is significantly more complicated. Although the above Equation (16) provides a precise solution for the pmf required, it is practically infeasible to compute even if the number n is of moderate size, as the set X_k contains $n!/[(n-k)!k!]$ elements. Fortunately, it is still possible to use probability generating function (pgf) to calculate even with a larger number of sources.

As in the model, in every time slot, each arrival process can be considered to satisfy a Bernoulli distribution, and because individual TCP traffic source is oblivious about any other sources even though they are connection through the same queue, each of the Bernoulli distribution describing each source can be considered independent. Let $X_i \sim Bernoulli(p_i)$, where $i = \{1, 2, 3, \cdots, n\}$ with the probability generating function (pdf) of

$$pgf = E[z^{X_i}] \tag{17}$$

As each of the distribution is Bernoulli, there is

$$pgf = E[z^{X_i}] = q + pz = (1-p) + pz \tag{18}$$

Let $S = \sum_{i=1}^{n} X_i$ denotes the sum of n such independent random arrival process, then, the pgf for the sum S of n can be described as

$$\begin{aligned} pgfS =& E[t^S] \\ =& E[t^{X_1}]E[t^{X_2}]E[t^{X_3}] \cdots E[t^{X_n}] \\ =& \prod_{i=1}^{n} [(1-p_i) + p_i z] \end{aligned} \tag{19}$$

The probability of k packets arriving in a time slot are given by the probability mass function (pmf), and according to the properties of pgf function, pmf can also be recovered by taking derivatives of pgf.

In this way, the probability that k packets arrives in a time slot can be calculated by

$$p_{ka} = \frac{G^k(0)}{k!} = \frac{d^k pgfS}{k! dz^k}|_{z=0} \tag{20}$$

This would produce the exact symbolic solution for the required probability as a function of the p_i. In the model, p_i is determined by the arrival rate of source i in that particular time slot, which is determined by other related factors such as congestion window size and their active transmission rate. Because of the relatively simple form of Bernoulli distribution pgf, it is not very computationally expensive as long as the number of source is manageable.

Thus, the state transition probability for queue length increase k from normal state is

$$\begin{aligned} p_i(k) =& p_{ka0d} + p_{(k+1)a1d} \\ =& (1-\beta)\frac{d^k pgfS}{k!dz^k}|_{z=0} \\ &+ \beta\frac{d^{(k+1)} pgfS}{(k+1)!dz^{(k+1)}}|_{z=0} \end{aligned} \tag{21}$$

As the model utilises the early arrival system (EAS) and centre observation mode, the state transitional probability starting from state 0 is

$$\begin{aligned} p_{i(k)first} =& p_{ka} = \frac{d^k pgfS}{k!dz^k}|_{z=0} \\ p_{r\,first} =& \prod_{i=1}^{n}(1-\alpha_i) \end{aligned} \tag{22}$$

Similarly, let the maximal queue capacity to be J, then the state transitional probability starting from state J is

$$\begin{aligned} p_{r\,last} =& (1-\beta) + \beta\frac{dpgfS}{dz}|_{z=0} \\ p_{d\,last} =& \beta\prod_{i=1}^{n}(1-\alpha_i) \end{aligned} \tag{23}$$

With the state transitional probability originating from every possible state defined, the state transitional matrix P can be easily generated as

$$P = \begin{bmatrix} p_{r\,first} & p_{i(1)first} & p_{i(2)first} & p_{i(2)first} & \cdots & p_{i(J-1)first} & p_{i(J)first} \\ p_d & p_r & p_{i(1)} & p_{i(2)} & \cdots & p_{i(J-2)} & p_{i(J-1)} \\ 0 & p_d & p_r & p_{i(1)} & \cdots & p_{i(J-3)} & p_{i(J-2)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & p_{d\,last} & p_{r\,last} \end{bmatrix} \tag{24}$$

This way, the state probability distribution of queue length during any successive time slot, and therefore, any performance measures derived from state probability distribution can be gathered using Equation (4).

It is also necessary to include some background traffic that is unresponsive to the dropping action performed in the queue, simulating traffic which do not normally control the transmission speed based on the dropping event in the network, such as that originated from a UDP source. This can be easily done by adding a traffic source that is always in "active transmission stage". A basic Geometric arrival process would satisfy this task with ease.

4 Evaluation for the Proposed Model

In this section, numerous numerical solutions with MATLAB and simulation experiments with NS-2 are performed. One particular scenario is selected as a demonstration of the proposed model and calculation method, while an instance of NS-2 simulation with equivalent parameters is performed as comparison. This is followed by more groups of data, gathered with both methods under different scenarios and parameters. The results are analysed in detail.

4.1 Detailed comparison in a demonstrative scenario

In most of the research utilising the conventional queueing theory and method, the results from the mathematical derivations remains theoretical. The parameters of the arrival and departure process are represented with probability and the performance measures are generally represented with the unit of time slots, which makes them very difficult to be compared with those from practical systems. One of the major reasons for that is with the discrete-time queueing model, the size of the time slot is not generally associated with the physical time unit. Most of the performance measures, with the exception of dropping probability, depends on the size of the time slot. The throughput and waiting time in discrete-time queueing model are both associated with the unit of time slot, which means that with the identical physical system, if different sizes of time slot are selected during two different modelling process, the throughput and waiting time value calculated would be different.

As the arrival rate α varied almost constantly, it would be more practical to define the size of the time slot base on the processing speed and bandwidth of the link leaving the gateway. An appropriate value can be assigned to β first, preferably a value between 0 and 1. In this way, the departure rate also represents the probability that a data packet leaves the queue within the time slot. The time represented by each time slot can thus be calculated by

$$T_s = \frac{\beta \times S_{pkt}}{B} \tag{25}$$

where T_s is the physical time represented by each time slot, S_{pkt} is the packet size, and B is the exiting bandwidth of the bottleneck gateway. Following this, a series of process of parameter conversion is performed, in order to map the variables used in the theoretical model and the parameters in the physical systems. These are specified as follow:

- Determine the physical time represented by each discrete time slot T_s by using the equation eq.(25).
- Calculate the total number of discrete time slot n_t in this experiment instance by

$$n_t = \frac{T_e}{T_s} \tag{26}$$

where T_e is the duration of the experiment. This can be used to determine if the experiment has reach the end.

- Calculate other factors for delay of the connection in unit of time slot D'_s by

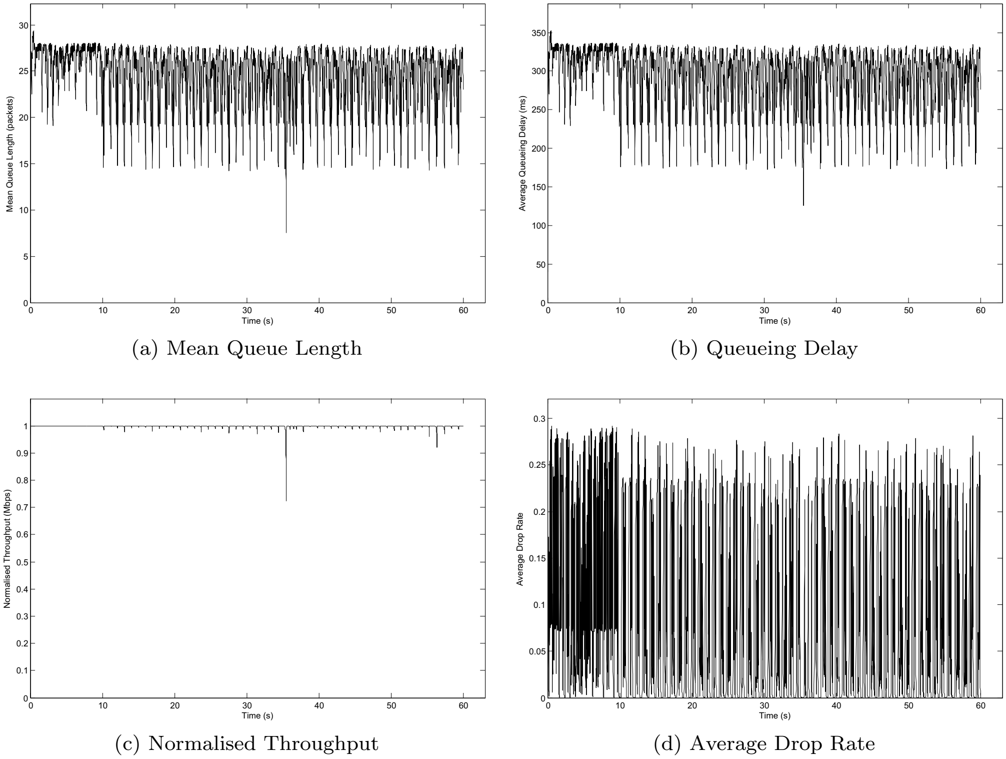
$$D'_s = \frac{D'}{T_s} \tag{27}$$

where D' is all other delays contributing the RTT. It is a parameter that needs to be specified before the experiment. This is combined with the queueing delay calculated for each packet and used to determine how long the acknowledgement can get back to the TCP source after the packet has been sent.

After this, it is possible to calculate the connection and the queueing behaviour of the bottleneck gateway recusively while capturing all the required performance measures. In a particular demonstrative scenario, parameters used by the MATLAB programme to perform numerical solutions are 4Mbps for the combining entering link bandwidth, which includes 18 TCP traffic sources of various active transmission rate. The sum of the active transmission rate of those TCP traffic sources are 3.6Mbps, with each of them valued from 0.12Mbps to 0.28Mbps with an increment of 0.01Mbps with another one valued 0.2Mbps (there are two sources with the value of 0.2Mbps). Beside from these, there is a background traffic of consistently 0.4Mbps transmitting through the queue, which simulates the unresponsive UDP-like sources. In the MATLAB numerical solution, this is performed by using a basic Geometrical arrival process with parameter corresponding to that bandwidth, and in NS-2 simulation, this is achieved by including a node which transmit constant bit rate (CBR) traffic with UDP protocol. The exiting link bandwidth remained at 1Mbps, as does the delay from all other factors other than the queueing delay in this particular queue at 200ms. The experiment duration is 60 seconds, the queueing discipline is first in first out with drop-tail queue and a maximal buffer capacity of 30 packets. The maximal congestion window sizes for all TCP sources are 65535 bytes, and all data packets are 1500 bytes in size. The size of data packets originated from the unresponsive UDP-like source is configured to be the same as those from normal TCP sources, so that they would not be distinguished at the queue based on their sizes. The initial transmission window size is 3000 bytes (2 packets).

With those parameters, the ratio between the responsive TCP traffic and unresponsive UDP traffic is 9:1, which is approximately the same as that in the practical network system [13]. The results of MATLAB numerical solution and corresponding NS-2 simulations are presented in Figures 1a to 1d and 2a to 2d. and the comparison of their average value is presented in Table 1. The NS-2 simulation results figures are result from a single instance, and in the table are the average of five different instances over their entire duration.

There are shape differences in the figures between two different methods, which is the result of single instance of simulation (NS-2) and a statistical calculation (MATLAB). As it can be observed from these results, in this scenario with multiple traffic sources, both responsive TCP-like source, and unresponsive UDP-like source,



(a) Mean Queue Length

(b) Queueing Delay

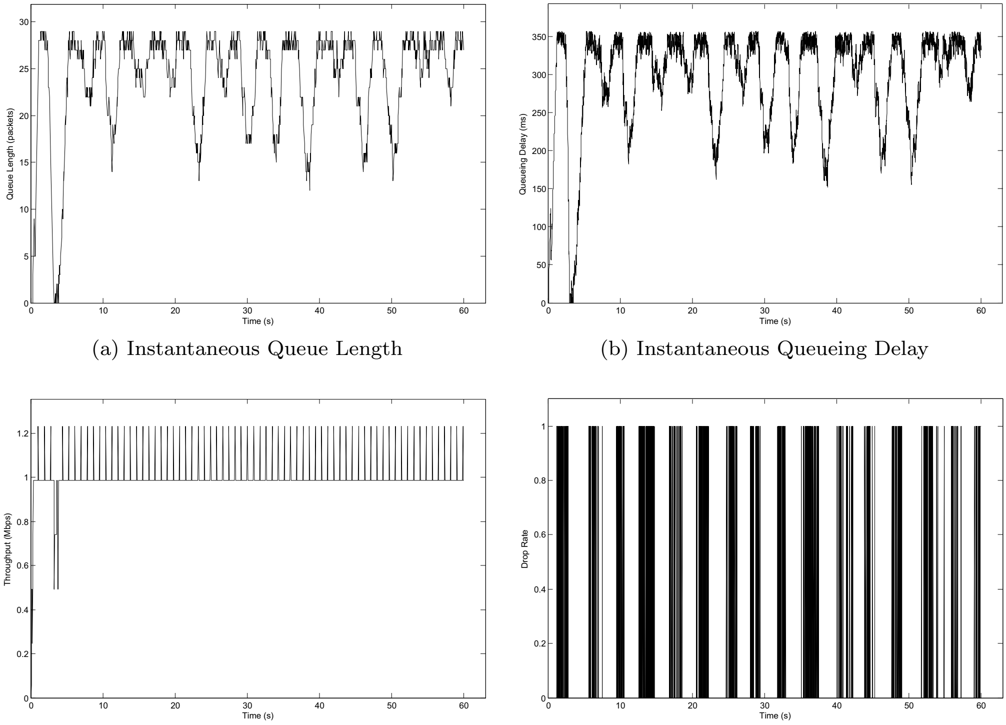
(c) Normalised Throughput

(d) Average Drop Rate

Fig. 1. Performance in MATLAB

the range at which queue length variant is not very significant. The presence of unresponsive UDP-like source ensures that the queue length and the throughput never decrease below the value that it alone can provide. With many TCP sources with similar specification sharing a same bottleneck queue which results in almost identical RTT for all of them, the congestion window size increase and decease speed for all of them are mostly the same, which would result in all responsive TCP-like source would increase and decrease their transmission rate at a similar rate. This would be a small scale "global synchronisation" phenomenon. From the perspective of the queue, the overall arrival rate would increase and decrease over the time slot at a much faster rate than that with only one or a few sources. As it is widely known, with a drop-tail queueing management scheme, it is impossible to avoid synchronisation between the traffic sources. This can be perfectly observed in this scenario, where the queue length and other related performance measures oscillate, increases and decreases many times in the short 60 seconds of experiments.

In the NS-2 figures, this can be clearly observed as the instantaneous queue length and resulting queueing delay increases to the higher point and decreases to a very lower level shortly after that. Similarly, in the MATLAB numerical solution figures, several points where the statistical value significantly lower than the average can be observed periodically. The throughput values gathered from both method indicate a very high utilisation percentage. Only certain points where queue length experience severe oscillation do the utilisation rate fall under 100%. This is because



(a) Instantaneous Queue Length

(b) Instantaneous Queueing Delay

(c) Throughput

(d) Instantaneous Drop Rate

Fig. 2. Performance in NS-2

the combined arrival rate is much higher than that of the departure rate, and at the same time, the queue length is sufficiently large, so that even with the arrival rate during some time period experiences heavy decrease, it would only drop to partially full, and rarely becomes empty. According to the characteristics of the single service queue, as long as the queue is not completely empty, the service would be fully utilised during that time. The constant background traffic which would single-handedly guaranteed a 50% link utilisation also greatly contribute to this.

In the average value comparison presented in the Table 1, the MATLAB numerical solution of the complex traffic model achieves a very close overall performance measures in all aspects comparing to the NS-2 simulation of the equivalent scenarios.

Method	Mean Queue Length (packets)	Average Queueing Delay (ms)	Average Throughput (Mbps)	Average Drop Rate
MATLAB	24.3658	292.5818	0.9991	0.077231
NS2	23.9551	295.7691	0.9940	0.072262

Table 1
Data for Complex Source

Buffer Size	20	30	40	50
Method	Mean Queue Length (packets)			
MATLAB	12.4266	24.3658	29.9699	36.7924
NS2	11.7380	23.9551	29.2252	35.9327
Method	Average Queueing Delay (ms)			
MATLAB	149.2167	292.5818	359.8756	441.7985
NS2	144.9269	295.7691	360.8383	443.6534
Method	Average Throughput (Mbps)			
MATLAB	0.9092	0.9991	0.9994	0.9997
NS2	0.9044	0.9940	0.9979	0.9985
Method	Average Drop Rate			
MATLAB	0.096539	0.077231	0.073370	0.067191
NS2	0.089605	0.072262	0.067926	0.061420

Table 2
Data for Various Buffer Size

4.2 Comparison with various parameters

In this subsection, more groups of MATLAB numerical solutions and NS-2 simulations with different parameters are performed. In each group, one MATLAB numerical solution will be calculated, as the model produces the statistical result. Several NS-2 simulations will also be performed in every group. The goal is to minimise the difference between the deterministic result of NS-2 simulation and the statistical result of MATLAB calculation. The experiments and simulations are divided into multiple groups, varying the buffer size (queueing capacity), overall arrival rate (entering link bandwidth), departure rate (exiting link bandwidth) one at a time. All the results from NS-2 simulations are the average value of five different instances.

4.2.1 Buffer size

In this group, the buffer size of the queue is changed between 20, 30, 40, 50, while all other parameters remain the same as in the previous demonstrative scenario. The results are presented in Table 2.

When the buffer size varies, queue length is the first to be affected. It would be intuitively obvious that when the maximal buffer size increases, so will the average queue length, as the dropping event will happen much later for the TCP sources. Higher queue length means larger queueing delay. However, as the queue are now less likely to be empty, the service utilisation rate also increases, and this is directly linked to the throughput, in which an increasing trend can also be observed. Overall, the results from simulation matches very well in all of the scenarios.

Arrival Rate	2	3	4	5	6
Method	Mean Queue Length (packets)				
MATLAB	23.1475	23.6348	24.3658	25.0968	25.5841
NS2	22.2782	22.9969	23.9551	24.9133	25.6320
Method	Average Queueing Delay (ms)				
MATLAB	277.9527	283.8043	292.5818	301.3593	307.2109
NS2	275.0653	283.9383	295.7691	307.5999	316.4729
Method	Average Throughput (Mbps)				
MATLAB	0.9691	0.9891	0.9991	0.9994	0.9994
NS2	0.9541	0.9741	0.9940	0.9558	0.9558
Method	Average Drop Rate				
MATLAB	0.073369	0.074914	0.077231	0.079548	0.081093
NS2	0.067204	0.069372	0.072262	0.075152	0.077320

Table 3
Data for Various Arrival Rate

4.2.2 Arrival rate

In this group, the effect of different arrival rates is analysed. The overall combined arrival rate (entering link bandwidth) of all the traffic sources is changed between 2Mbps, 3Mbps, 4Mbps, 5Mbps and 6Mbps, with bandwidth ratio of the traffic sources unmodified. All other parameters remains as that in the demonstrative scenario. The results are presented in Table 3.

With the increasing overall arrival rate, a steady increasing trend of queue length can be observed, as well as other associated performance measures, such as queueing delay and throughput. The dropping rate also increases as a direct result of heavier traffic load on the bottleneck gateway. Overall, the two groups of data match quite well with each other. At higher overall arrival rate, the traffic sources simply have to spend a larger ratio of time in its “idle stage” instead of “active transmission stage’. Ultimately, the congestion window size increasing speed depends on RTT, as during congestion avoidance stage, window size can only increase one MSS for every RTT. According to the analysis performed in section 3, the congestion window size is the deciding factor for the transmission speed, which is the arrival rate of the queue. As the combined RTT from other sources is the same for all experiments, the overall RTT is also going to be very similar due to the similarity of queue length in all higher rate scenarios.

4.2.3 Departure rate

In this final group of experiments, the service rate (exiting link bandwidth) is changed between 0.4Mbps, 0.7Mbps, 1Mbps, 1.3Mbps and 1.6Mbps, while all other parameters remain the same as the previous demonstrative scenario. The results are represented in Table 4.

With the increasing departure rate, and generally decreasing trend of queue length, as well as other associated performance measures, namely, queueing delay

Departure Rate	0.4	0.7	1	1.3	1.6
Method	Mean Queue Length (packets)				
MATLAB	26.1932	25.4623	24.3658	24.0003	23.5130
NS2	26.2308	25.2726	23.9551	23.3562	22.6376
Method	Average Queueing Delay (ms)				
MATLAB	314.5254	305.7480	292.5818	288.1931	282.3414
NS2	323.8672	312.0364	295.7691	288.3749	279.5018
Method	Average Throughput (Mbps)				
MATLAB	0.9993	0.9991	0.9991	0.9841	0.9641
NS2	0.9958	0.9958	0.9940	0.9691	0.9392
Method	Average Drop Rate				
MATLAB	0.083023	0.080706	0.077231	0.076073	0.074528
NS2	0.079126	0.076236	0.072262	0.070455	0.068288

Table 4
Data for Various Departure Rate

and throughput. The dropping rate also decreases slightly. There is a similar trend in reverse that was observed in the increasing overall arrival rate scenarios. It can be concluded that the values from numerical solutions match well with those from NS-2 simulations. This, along with two previous groups of scenarios with various buffer sizes and various arrival rates, provide potent validation for the proposed modification of conventional queueing method and dynamic traffic source model.

5 Conclusion and Future Work

In this paper, the limitation of the conventional method is analysed and a new interactive model of multiple dynamic responding traffic sources with a time-variant heterogeneous queue is presented. The proposed model utilises the time-variant queueing calculation method, and no longer requires the assumption of arrival rate strictly smaller than departure rate. In comparison, the conventional method usually ignores all stages of flows where arrival rate may be larger than the departure rate, which is not a rare occasion in practical network systems. The proposed model is validated extensively by performing various groups of MATLAB numerical solutions with different parameters and comparing each of them with NS-2 simulation experiments. The results have shown close match between them, which proves the effectiveness and accuracy of the model.

Drop-tail queues are used in the analysis of this reach. However, this can be easily substituted with any particular AQM scheme, and the time-variant modelling process has already provided an interface for the incorporation of the active dropping mechanism and dynamic dropping criteria decision.

The proposed model calculates the state probability distribution recursively, and provides performance measure statistics for the transient period of the queue. This could prove useful for the performance evaluation of system with heavily fluctuating traffic and oscillating queue length behaviour which consists larger portion of transient activity. Further extension and improvement to the proposed model would be performed to alleviate the increase of computational cost due to the recursive calculation nature of the model.

References

[1] *Rfc: 793, transmission control protocol*, http://www.ietf.org/rfc/rfc793.txt (1981).

[2] *Rfc: 6582, the newreno modification to TCP's fast recovery algorithm*, http://www.rfc-editor.org/rfc/rfc6582.txt (2012).

[3] Artalejo, J. and O. Hernndez-Lerma, *Performance analysis and optimal control of the Geo/Geo/c queue*, Performance Evaluation (2003), pp. 15–39.

[4] Asmussen, S. and J. L. Teugels, *Convergence rates for M/G/1 queues and ruin problems with heavy tails*, Journal of applied probability (1996), pp. 1181–1190.

[5] Atencia, I. and P. Moreno, *The discrete-time Geo/Geo/1 queue with negative customers and disasters*, Computers and Operations Research (2004), pp. 1537–1548.

[6] Bansal, D., H. Balakrishnan, S. Floyd and S. Shenker, *Dynamic behavior of slowly-responsive congestion control algorithm*, in: *SIGCOMM*, 2001.

[7] Chen, Z., L. Guan and P. Bull, *Dynamic arrival process for TCP using a novel time-dependent statistical calculation approach,*, in: *2014 IEEE International Conference on Future Internet of Things and Cloud*, IEEE, 2014, pp. 27–29.

[8] Dattatreya, G. R., "Performance Analysis of Queueing and Computer Networks," CRC Press, 2008.

[9] Fuhrmann, S. W. and R. B. Cooper, *Stochastic decompositions in the M/G/1 queue with generalized vacations*, Operations Research **33** (1985), pp. 1117–1129.

[10] Granovsky, B. L. and A. Zeifman, *Nonstationary queues: Estimation of the rate of convergence*, Queueing Systems **46** (2004), pp. 363–388.

[11] Gross, D. and D. R. Miller, *The randomization technique as a modeling tool and solution procedure for transient Markov processes*, Operations Research **32** (1984), pp. 343–361.

[12] Hassan, M. and R. Jain, "High Performance TCP/IP Networking," Pearson Education, 2004.

[13] Hilbert, M. and P. Lopez, *The world's technological capacity to store, communicate and compute information*, Science **332(6025)** (2011), pp. 60–65.

[14] Katabi, D., M. Handley and C. Rohrs, *Congestion control for high bandwidth-delay product networks*, in: *SIGCOMM*, 2002, pp. 89–102.

[15] Kelton, W. D. and A. M. Law, *The transient behavior of the M/M/s queue, with implications for steady-state simulation*, Operations Research **33** (1985), pp. 378–396.

[16] Kumar, B. K. and D. Arivudainambi, *Transient solution of an M/M/1 queue with catastrophes*, Computers & Mathematics with applications **40** (2000), pp. 1233–1240.

[17] Leguesdron, P., J. Pellaumail, G. Rubino and B. Sericola, *Transient analysis of the M/M/1 queue*, Advances in Applied Probability **25** (1993), pp. 702–713.

[18] Li, J. H. and N. S. Tian, *Analysis of the discrete time Geo/Geo/1 queue with single working vacation*, Quality Technology and Quantitative Management (2008), pp. 77–89.

[19] Misra, V., W.-B. Gong and D. Towsley, *Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED*, in: *ACM SIGCOMM*, 2000, pp. 151–160.

[20] Mohammed, A. M. and A. F. Agamy, *A survey on the common network traffic sources models*, International Journal of Computer Networks (IJCN) (2011), pp. 103–115.

[21] Muscariello, L., M. Mellia, M. Meo, M. A. Marsan and R. L. Cigno, *Markov models of internet traffic and a new hierarchical mmpp model*, Computer Communications (2005), pp. 1835–1851.

[22] Olawoyin, L., N. Faruk and L. Akanbi, *Queue management in network performance analysis*, International Journal of Science and Technology (2011).

[23] Posey, B. M., "Networking Basics: Part 1 - Networking Hardware," TechGenix Ltd, 2006.

[24] Stidham, S., *Analysis, design, and control of queueing systems*, Operations Research **50** (2002), pp. 197–216.

[25] Szigeti, T. and C. Hattingh, "End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs," Cisco Press, 2004.

[26] van Foreest, N., M. Mandjes and W. Scheinhardt, *Analysis of a feedback fluid model for heterogeneous TCP sources*, Stochastic Models (2003), pp. 299–324.

[27] Wadsworth, G. P., "Introduction to probability and random variables," McGraw-Hill, 1960.

[28] Wang, Y. H., *On the number of successes in independent trials*, Statistica Sinica (1993), pp. 295–312.

[29] Woodward, M. E., "Communication and computer networks: modelling with discrete-time queues," London Pentech, 1993.

[30] Worthington, D. and A. Wall, *Using the discrete time modelling approach to evaluate the time-dependent behaviour of queueing systems*, Journal of the Operational Research Society (1999), pp. 777–788.

[31] Xiong, Y. and Z. Bu, *Density evolution method for discrete feedback fractal M/M/1 traffic models*, in: *2005 International Conference on Communications, Circuits and Systems*, 2005.