HYBRID MODELING OF THE DYNAMIC BEHAVIOR OF MOBILE AD-HOC NETWORKS

by

Siriluck Tipmongkonsilp

B.Eng.,Chulalongkorn University, Thailand 1997

M.S., Electrical Engineering, Carnegie Mellon University, 2001

Submitted to the Graduate Faculty of

the School of Information Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2009

UNIVERSITY OF PITTSBURGH

SCHOOL OF INFORMATION SCIENCES

This dissertation was presented

by

Siriluck Tipmongkonsilp

It was defended on

August 14, 2009

and approved by

David Tipper, Ph.D., Associate Professor, School of Information Science

Yi Qian, Ph.D., National Institute of Standards & Technology

Richard Thompson, Ph.D., Professor, School of Information Sciences

Prashant Krishnamurthy, Ph.D., Associate Professor, School of Information Sciences

Joseph Kabara, Ph.D., Assistant Professor, School of Information Sciences

Taieb Znati, Ph.D., Professor, Department od Computer Science

Dissertation Director: David Tipper, Ph.D., Associate Professor, School of Information Science

HYBRID MODELING OF THE DYNAMIC BEHAVIOR OF MOBILE AD-HOC NETWORKS

Siriluck Tipmongkonsilp, PhD

University of Pittsburgh, 2009

The performance of mobile ad-hoc networks is normally studied via simulation over a fixed time horizon using a steady-state type of statistical analysis procedure. However, due to the dynamic nature of the network topology such an approach may be inappropriate in many cases as the network may spend most of the time in a transient or nonstationary state. The objective of this dissertation is to develop a performance modeling framework and detailed techniques for analyzing the time varying performance of mobile ad-hoc networks.

Our approach is a performance modeling tool for queueing analysis using a hybrid of discrete event simulation and numerical method techniques. Network queues are modeled using fluid-flow based differential equations which can be solved with any standard numerical integration methods, while node connectivity that represents topology changes is incorporated into the model using either discrete event simulation techniques or stochastic modeling of adjacency matrix elements. The hybrid fluid-based approach is believed to be an alternative that can resolve certain issues in current simulators and provide flexibility in modeling a more complex network by integrating additional features of nonstationary effect to add higher level of fidelity into the proposed model. Numerical and simulation experiments show that the new approach can provide reasonably accurate results without sacrificing a large amount of computational resources.

To my parents, Tawatchai and Nilubol Tipmongkonsilp

and my husband, Dr. Boonchuan Immaraporn

ACKNOWLEDGEMENT

This work would not have been possible without the guidance of my advisor and committee members, as well as the support from my parents and my family.

I would like to take this opportunity to express my gratitude to Dr.Tipper, my advisor, for his guidance, encouragement, inspiration, and financial support through out the course of my research. I also would like to thank Dr.Qian, my co-advisor, for all of his valuable comments and suggestions.

Most importantly, I would like to thank my parents and my husband. I deeply appreciate the love and support they have for me. This work is dedicated to you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

1.1 BACKGROUND

Mobile ad-hoc networks (MANETs) are expected to become a major component of the communication infrastructure in the future. MANETs are comprised of mobile nodes which can dynamically self organize into arbitrary temporary "ad-hoc" topologies, allowing users and devices to seamlessly network without a pre-existing communication infrastructure. The mobile nodes must cooperate to dynamically establish routes using wireless links, and routes may involve multiple hops with each node acting as a router. Since the mobile network nodes can move arbitrarily, the network topology is expected to change often and unpredictably. Hence, ad-hoc networks require highly adaptive protocols and efficient failure recovery strategies to deal with the frequent topology changes. MANETs also inherit the traditional problems of wireless communications and networking (e.g., broadcast communication channels, asymmetric channels and signal propagation, energy constraints in mobile nodes, links that are poor quality in comparison to wired links, hidden terminal and exposed terminal problems, etc.), which when combined with the unique mobility and lack of infrastructure features make their design and development challenging [1].

Fundamental to the design of efficient MANETs is the ability to estimate and predict the network performance. Traditionally, mobile ad-hoc network performance has been evaluated using discrete event simulations. The basic approach adopted in most literature for a given network scenario is to conduct a simulation under the simplifying approximation of steady-state conditions over a fixed period of time, with multiple runs from different random number seeds, and collected data being averaged over the runs. It is recommended

that the observation of performance measures during the initial transient period is not a good representative of the steady state behavior and statistics with high variabilities gathered during the transient period in each run are usually eliminated to avoid initialization bias. Such approach is considered as a steady state simulation methodology, termed the replication/deletion approach. However, the control strategies obtained from steady state simulation may only be optimal under stationary conditions and impractical in MANETs, since nonstationary conditions may exist and significantly impact the performance of the network [2], [3]. The nature of MANETs suggests that transient conditions are likely to occur and possibly dominate the network behavior, as a result time varying behavior becomes important and meaningful. Simulation study of the time varying behavior for such networks is possible, though computationally difficult. To study nonstationary networks, the measurements of quantities observed over small intervals or at specific points in time are important, therefore time averages are not the proper approach and ensemble averages are more appropriate. The idea is to construct ensemble average curves of quantities of interest across a set of statistically identical but distinct independent simulation runs, along with the calculated confidence interval. With many such points collected at different time instants, behavior of the system can be shown as a function of time. However, to assure the accurate portrayal of the actual system, a large number of runs is required resulting in large amount of CPU time. This is computational expensive, and difficult to scale even with the advent of specialized simulation tools such as GloMoSim [4], OPNET [5], ns-2 with CMU extension [6], or QualNet [7] to reduce the complexity in the simulation modeling.

A recent survey article on discrete simulators [8] shows that, though offering convenience in modeling advanced simulation environments, there are common simulation shortfalls often overlooked in these tools, including model validation, data initialization, repeatability issues, metric collection, and statistical analysis that one way or another have effects on the performance results. In [9] and [10], some concern has been raised when the resulting performance from ns-2, GloMoSim and OPNET Modeler, being compared to one another, have shown some differences between the simulators. The importance of effects of detail in network simulation has been emphasize yet again in [10]. Realistic simulation infused with

detail may provide accurate results; however too much detail leads to slow and cumbersome simulators. On the other hand, simulation that lack necessary elements may return misleading and incorrect results.

Network performance evaluation can also be achieved by defining the model using analytical techniques. The standard analytical network model used in network performance evaluation mainly deals with steady state conditions by using queueing theory techniques. The time varying analysis involves solving a set of differential equations, and it is generally difficult to obtain the solution in closed form expression even for a simple scenario [2], [11], [12]. To make the analytical technique more manageable, it typically requires a number of simplifying assumptions to get a tractable model.

A weakness of most of the literature on the performance of mobile ad-hoc networks is that steady state analysis techniques are used even though transient or nonstationary periods will occur in the network, especially after a link or node failure. The importance of this transient behavior after link/node failures has been illustrated in several network technologies including circuit switched networks [13], packet switched data networks [14], packet based signaling networks [15], digital cellular wireless networks [16], [17] and recently in MANETs [2]. This work, taken together, demonstrates that the dominant factor on network performance after a dynamic event such as a link failure is the transient or nonstationary congestion period. Due to the mobility of nodes and their limited battery life, link and node failures are common in MANETs. Thus, one would expect that transient/nonstationary conditions to occur often and likely dominate the performance behaviour [2]. Hence, routing schemes, QoS mechanisms, and congestion control techniques designed and evaluated via steady state analysis may not make optimum use of network resources after a failure or during nonstationary periods. In summary, while significant progress has been made towards developing mobile ad-hoc network protocols [1] and developing simulation tools [4], [5], [6], [7] to estimate their performance, relatively little work has appeared on performance models of mobile ad-hoc networks which capture their time varying behavior.

1.2 PROBLEM STATEMENT

The main motivation of this research comes from the concerns of the drawbacks of existing simulation tools currently used to study the performance of networks with the presence of time-varying conditions:

- the overly simplifying assumptions that lead to invalid simulation models,
- user errors from not following correct simulation methodology, and
- doubts on accuracy and scability on simulators

The objective of this dissertation is to develop a new performance modeling technique that focuses on the dynamics of MANETs, which includes both time varying and steady state behavior. Such performance modeling technique should analyze the time varying behavior of network queues in MANETs, while in the same time offer efficiency, scalability and high fidelity. The proposed performance modeling technique is a hybrid of two elements: discrete event simulation and numerical method based queueing analysis. While network queues are modeled using fluid flow based differential equations, which can be solved by using standard numerical integration methods, node connectivity representing topology changes is integrated into the model using either discrete event simulation or numerical stochastic based modeling techniques. The proposed fluid flow approximation model is believed to be a flexible alternative to represent network queue undergoing nonstationary conditions within a reasonable degree of accuracy. Such an approach is more scalable than a standard non-stationary discrete event simulation software and allows the insight into the interaction of network nodes/protocols.

1.3 DISSERTATION OUTLINE

The rest of the dissertation is organized as follows: Chapter 2 provides a background of available literature on network performance evaluation tools and general time varying behavior

modeling. Chapter 3 presents the details of our hybrid modeling approach, which consists of the network topology modeling and the nonstationary queueing modeling. Chapter 4 explores other possible network performance measures related to the fluid model that can be used to assess the time varying behavior of the network, followed by the numerical experiments in Chapter 5. Finally, Chapter 6 concludes this dissertation and discusses directions for future research.

2.0 LITERATURE REVIEW

2.1 NETWORK PERFORMANCE EVALUATION TOOLS

Network performance can generally be evaluated using either measurements, simulation or analytical models [18]. Measurement techniques involve gathering measurement data from experiments running on the real systems or working testbeds that represent the network of interest. Simulation methods are computer-based modeling over a range of network scenarios and parameter values, with the performance metrics being determined from a simulation. The third one, analytical models determine the performance using mathematical analysis techniques.

2.1.1 Measurement-based Performance Evaluation Technique

Since measurement studies need to construct a real ad hoc network testbed for a given scenario to measure network performance and collect trace data, they are typically expensive and possible only for limited working scenarios. The measurement-based techniques also come with several major drawbacks, including the difficulty in generalizing results, replicating the measurement environments and the effort required for considering all possible cases. There are very few measurement-based studies on MANETs found in the literature so far [19]. Despite the difficulty in constructing a real working ad hoc testbed, measurement-based network performance evaluation studies can reveal some useful discoveries that might be hidden in simulation or analytical-based methods. For example, in [19], the results from the Uppsala University APE testbeds pointed out problems of *communication gray zones* where actual data communication cannot get through even when routing table indicates ex-

isting valid routes and neighboring nodes should be reachable, resulting in a large number of unexplained packet losses. The magnitude of packet losses is larger than what can be explained by the re-routing that normally occurs when a node losses its contact with its next hop. The problem was discovered while a group of researchers at Uppsala University were measuring the performance of AODV routing protocol in an ad hoc network implementation.

In [20]-[21], an outdoor experiment was conducted to examine the integrity of commonly used assumptions in wireless simulations, which are flat earth, omnidirectional radio propagation, equal radio range, symmetric communication or perfect radio propagation. The results of the outdoor experiment were compared with the results obtained from the best-effort simulation model that only employed some of the assumptions such as flat earth, omnidirectional radio propagation and symmetric communication. Then the simulation model was weakened with additional axiom assumptions to see how the results would be impacted quantitatively. Since each axiom assumption can be contradicted both by intuition and by experimental results, the outcomes from simulation model were expected to deviate from the reality. However in the case of best-effort model, the simulation produced good results that were reasonably matched to their particular outdoor experiment scenario. The paper suggested that the axiom assumptions are undoubtedly invalid in many situations; however, it is possible for the realistic stochastic model that carefully describes the chosen target environment to return acceptable results in the context of those conditions and assumptions.

Another example of wireless mesh measurement testbed is at the Quail Ridge Natural Reserve, called *QuRiNet* [22]-[23]. The network is originally used for environmental research study of the flora and fauna in the region by the Department of Ecology at UC Davis. This QuRiNet claims to be different from other measurement-based simulation in terms of its location and its usage, so that the validity of theoretical ideas under practical situation can be tested, when complicated issues such as hilly terrain, forest growth, or long distance are involved. Preliminary results on the network utilization, round trip time performance, signal strength and network capacity were provided. Due to geographical factors, they found out that the same setting from indoor laboratory testbed did not work on QuRiNet and the

parameters such as the power level of wireless cards, the antenna type, or the placement of antennas should be adjusted accordingly. One observation from the project was asymmetric signal strengths and throughput values between the same two nodes. Even on the same link, there are large variations on both signal strength and throughput at different points of time during a day. It is however important to point out that many communication problems observed from measurement testbeds cannot be easily modeled by commonly used simulation tools.

2.1.2 Simulation-based Performance Evaluation Technique

The simulation studies, unlike measurement-based approach, only deal with computer modeling over a range of network scenarios and parameter values. Unfortunately, simulation still shares the same drawbacks as in measurement, specifically trouble in generalizing results and the effort involved in considering all cases. The performance measures determined from a simulation are random in exactly the same manners as measurements; hence confidence intervals must be used when discussing performance measures estimated by simulations. To assure independently and identically distributed data generated in simulation studies and increase credibility of statistical simulation results [24], large amount of computer run time for independent multiple runs with confidence intervals on results are required.

Despite this concern, simulation techniques are still popular tools of choice in network performance evaluation because they allow the network to be modeled to an arbitrary degree of accuracy and detail. Most computer-based simulation studies are based on using existing simulation tools rather than developing new software. Popular computer-based discrete-event network simulation tools include GloMoSim [4], OPNET [5], ns-2 [6], and Qualnet [7]. These tools construct a simulator that mimics system state transitions, collects the system data during simulation runs, and then analyzes the collected data to estimate the performance metrics. They all provide convenience in network modeling with libraries containing predefined models as well as offer advanced simulation environments to test and debug dif-

ferent network protocols. However, the validity of MANET simulations based on these tools has been questioned in recent literature. For example, in [10], the differences between the simulation results of simple flooding algorithm using GloMoSim, OPNET, and ns-2 were measured both quantitatively and qualitativly, while in [9], the level of detail in configuring and implementing the simulation scenarios also leads to divergences between results from these simulator tools.

Related to simulation models in recent research studies, an excellent discussion of the modeling issues in building more accurate models is given in [24], [25]. The major issues in design and modeling of MANETs include the geographic space in which the mobiles move, geographic boundary policy, number of nodes and their position distribution, signal propagation models, signal interference models, mobility models, and the traffic workload characteristics. To construct a model that matches the goal of the simulation project, all issues should be carefully considered to sufficiently include relevant elements that capture realistic effects and prevent any nonnecessary detail that will introduce overheads and delay in simulation process. Since mobility plays a central role in the MANET model design, much of the effort in the literature has focused on the mobility and boundary policy. Typical simulation studies assume fixed number of active mobile nodes implemented in a closed two-dimensional limited area of interest with some bouncing policies defined for users arriving at the edge of the simulated area.

Most of the simulation tools model the radio propagation of MANETs using simplifying assumptions that all radios have a perfect coverage on a two-dimensional circular range, even with the awareness of the necessity and complexity of realistic radio propagation models. The set of fundamental axioms of upon which most MANET research relies, includes flat earth model, circular transmission, equal range radio, symmetric communication, perfect coverage, and signal strength as a simple function of distance. From the survey of Mobicom and Mobihoc proceeding from 1995 through 2002 [20]-[21], the great majority of papers have embodied the combination of these axioms for radio propagation modeling and produced simulation results for MANETs. Basically, in flat earth radio models, two nodes can com-

municate if one is simply within some distance of the other. On the other hand, the simple radio model, commonly used in simulation research, are able offer a more realistic detail (i.e., packet delay model, interference and shadowing effect) than just present or absent as in flat earth models. In [20]-[21], it has been pointed out that the data from real world experiment strongly contradicts these assumptions and raises the questions on published simulation results.

These simulators ensure the strict chronological order of the simulation by processing one event at a time; for this reason, they are called sequential simulators. However, one major drawback of such simulators is that they take long time to run the process even for steady state behavior, which leads to execution time issues and scalability problems. Several simulation techniques have been proposed to speed up the execution of simulation of large-scale wireless networks. Parallel simulation and distributed simulation approaches exploit the idea of concurrency among events to achieve a reduction in execution time. Alternatively, using federated simulation where separated subnetworks of the simulated model are executed on different processors has been proposed [25], [26]. Researchers have also tried to apply the concept of parallel and distributed simulation techniques to the development of a more realistic simulation tool for nonstationary network behavior, however the scalability is still believed an open issue.

2.1.3 Analytical-based Performance Evaluation Technique

Network performance evaluation can also be achieved by defining a system model and solving the model using analytical techniques. The standard analytical network model used in network performance evaluation mainly deals with the steady state conditions of queueing theory models. Performance metrics must be carefully defined to evaluate and understand the critical features of the considered system, since some metrics such as delay, throughput, or utilization, are general and can be used to represent many protocols while others may be relevant for a given specific protocol layer [25]. Steady state performance metrics are typically in terms of basic measured quantities that are consistent with measurements made

on a physical system such as flow or throughput, number of data units stored in devices, point-to-point delay, or utilization. In the analysis of the metrics, these quantities are normally time-averaged to obtain results with some generality [27].

However, to model the realistic dynamics of time-varying behavior when parameters are not constant anymore, a stochastic process is commonly used to describe the network behavior under nonstationary conditions. The analysis rather involves solving a set of differential equations, which is generally difficult to obtain the solution and rarely in the closed form expression even for a simple scenario. To make the analytical technique more manageable, they typically require a number of simplifying assumptions to get a tractable model.

2.2 NONSTATIONARY BEHAVIOR MODELING

Most of the literature on network performance from the previous section focuses on steady state modeling, while there are not many widely available works on nonstationary characteristics of the networks. One intuitive approach for dealing with nonstationary conditions is to partition time into segments where steady state conditions can be assumed, and steady state results can be applied. However, to use steady state performance measures to represent a simple system undergoing nonstationary conditions and capture the time-dependent performance, such results need to be modified.

2.2.1 Approximation Methods for Nonstationary Behavior

The techniques for stationary approximation have been introduced in [28], [29] in the context of a simple Markovian queueing model with nonstationary arrivals. It reflects the idea of using steady-state analysis at each instant of time while fixing the arrival rate to be constant. There are four different approximation methods, namely the simple stationary approximation (SSA), the peak approximation (PA), the Quasi-Static Approximation, and the Pointwise Stationary Approximation (PSA).

The simple stationary approximation (SSA) method replaces time-varying parameter with its steady state mean value. The SSA approximation method has a long history in the literature for the reason that it is simple and applicable to a wide range of nonstationary queueing system. The numerical results in [30] show that the SSA method can provide reasonably accurate results only in systems with relatively low variation in nonstationarity; however, the SSA will behave poorly when applied on highly nonstationary systems. In general, the SSA underestimates the average performance measure of nonstationary systems as it only provides the average performance measures of the nonstationary system as oppose to a time-dependent measure.

Similar to the SSA method, the peak approximation (PA) technique uses the computation of average performance measures of the systems during the peak period, instead of the average. The idea of dealing with peak performance is quite common in telecommunication applications where busy hour performance is used as an upper bound metric [31]. By considering only the maximum value, the PA technique is deemed as an extreme case of averaging out the nonstationary conditions.

The Quasi-Static Approximation (QSA) approach, sometime referred to as quasistationary, monitors the time-varying parameters at a set of periodic intervals and re-calculates the steady state at each period. To allow the network to attain steady state and use steady-state results for each time interval, this approach assumes static conditions during each instant in time.

Lastly, the Pointwise Stationary Approximation (PSA) is comparable to the QSA, except that in this case, the parameters are now sampled at various time points with steady state results being used at each time point. This method is based on the idea that the nonstationary condition approximately behaves like a stationary model at each time instance, thus the steady-state result can then be used to approximate the nonstationary queueing system at each point on time. The PSA method can be easily generated to most of the queueing

systems. The accuracy of these techniques depends greatly on the rate and magnitude of time-varying changes. In [11], the simulation results using PSA on different queueing systems with various traffic patterns were presented; however, in some cases, the simple PSA was not enough to represent network dynamics and returned poor results. Similarly to the stationary systems that is stable when $\rho < 1$, when applying PSA approach to nonstationary system, one has to make sure that $\rho(t) < 1$ for all t to guarantee the system stability [32], [33].

2.2.2 Overview of Nonstationary Queueing Models

The words *transient* and *nonstationary* have been used interchangably to describe time-varying behavior in networks; however there is in fact an distinction between transient behavior and nonstationary behavior. Transient behavior refers to the system going from one stationary state to another, whereas nonstationary behavior applies to the specific case of continuous variation in arrival rate or service rate. In transient behavior studies, typical performance metrics of interest are relaxation time or time to reach steady state, mean behavior, or distributional behavior. The property of the system under study as well as the purpose of the study will determine which performance measure best describes the system behavior.

Most of the literature of queueing network analysis focuses on the stationary conditions. When there are changes in traffic demand or network topology applied to the system, a certain degree of adaptivity in routing and network control parameters must be recalculated, while assuming static conditions during each updating period so that networks can attain steady state conditions again. There have been works to develop closed form solutions that capture the time varying dynamics of the queue; however even the simple queue such as a system with Poisson arrival and exponential packet size ($M/M/1$) with known closed form transient solution, it is notoriously difficult to solve [12]. Hence, there is an emphasis on numerical techniques to solve for transient and nonstationary behavior.

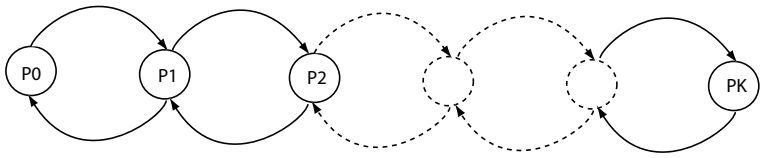

Figure 2.1: State Transition Diagram of a Single Queue

As an example, one early analytical based numerical scheme describes the network time dependent behavior of a finite capacity embedded Markovian queue, shown in Figure 2.1, in terms of time varying probability distributions using the *Chapman-Kolmogorov* forward (C-K) differential equations as in Equation (2.1). Here $p^i(t)$ is the probability of i customer in the single queueing system at time t, $\lambda^n(t)$ is the average arrival rate to the queue if there are n customers in the system at time t, $\mu^n(t)$ is the average service rate if there are n customers in the system at time t, and K is the capacity of the system. From Equation (2.1), to study a system queue size K, $(K+1)$ differential equations are needed. This large set of differential equations is computationally complex to evaluate, thus settling with solution for transient behavior numerically is easier than deriving a closed form expression.

$$\begin{aligned}
\frac{dp^0(t)}{dt} &= -\lambda^0(t)p^0(t) + \mu^1(t)p^1(t) \\
\frac{dp^n(t)}{dt} &= \lambda^{n-1}(t)p^{n-1}(t) - \lambda^n(t)p^n(t) - \mu^n(t)p^n(t) + \mu^{n+1}(t)p^{n+1}(t) \\
&= \lambda^{n-1}(t)p^{n-1}(t) - [\lambda^n(t) + \mu^n(t)]p^n(t) + \mu^{n+1}(t)p^{n+1}(t) \qquad 0 < n < K \\
\frac{dp^K(t)}{dt} &= \lambda^{K-1}(t)p^{K-1}(t) - \mu^K(t)p^K(t) \qquad (2.1)
\end{aligned}$$

This set of differential equations is generally difficult to solve by any analytical means; however, in [34], they study the transient behavior of a stationary $M/G/1$ network queue with the basic idea of approximating the time-varying average arrival and service rates to be constant over small time intervals so that steady-state equilibrium can be reached and solution of these equation can eventually be obtained. Such approximation allows the possibility of solving these set of differential equations for the state probabilities over each time interval. Start with the initial $p^n(t_0)$, the system state at the end of the first time interval

$[t_0, t_1]$, given by the probability distribution $p^n(t_1)$ becomes the boundary conditions for the next time interval $[t_1, t_2]$, and the procedure is repeated for the all remaining time intervals. The result of the analysis is a transform expression for $p^i(t)$, the time varying probability of i customers in a single queueing system. Though the time dependent state probability is useful for determining a numerical evaluation of the network performance for a given time varying arrival and service pattern during both nonstationary and steady state periods, the result is usually computationally complex to evaluate. Since general routing and flow control parameters are commonly based on optimization and feedback of average quantities, some possible network performance measures are $L(t)$ - the expected number of customers in the system at time t, $D(t)$ - the departure rate from the queue at time t, $RR(t)$ - the rate of packets being rejected by the network, or $TN(t)$ - the total network link flow at time t.

To address the problem of computational complexity when solving the for the time varying probability $p^i(t)$, a numerical approach called randomization or uniformization analysis techniques was proposed [35]. The basic idea is to decompose Continuous Time Markov Chain (CTMC) into discrete time Markov chain and truncated Poisson process in order to study transient behavior, while numerical analysis techniques solve underlying differential/difference equations with standard numerical analysis techniques. Both the C-K differential equation method and uniformization analysis method are primarily limited to Markovian systems and become more computationally complicated as queue state space grows, the two numerical methods provide reasonably accurate results for typical performance measures. To study a more general queueing systems, a number of approximate analysis methods, such as diffusion models, fluid flow models, and service time convolution, has been proposed to provide ease of computation and offer accurate results within reasonable bounds.

This dissertation is more focus on the mean nonstationary behavior of queueing system using an approximate fluid flow modeling method, which was introduced in [2] and [11]. The technique reflects the blend of two concepts: fluid modeling and pointwise stationary approximation. While the fluid flow modeling refers to the network queue with general arrival and service process that are characterized by flow in and flow out rate, the term pointwise

stationary approximation (PSA) adopts the idea of using steady state analysis at each time instant and computes for the performance measure at particular time point during the period of interest. The fluid model describes the dynamics of the network with a set of nonlinear differential equations in terms of the time varying quantities of performance metrics such as queue length, on which optimization of control strategies are commonly based. Such time dependent differential equations are nonlinear and generally difficult to solve. However, with the help of PSA concept, it is possible to numerically solve a set of nonlinear differential equations over each time interval and repeat the procedure over the entire time interval using any standard integration routines [2], [12], [30], [11]. To understand how the fluid flow model coupled with the pointwise stationarity can be used to analyze the network performance as well as determine the mean nonstationary behavior of the network, we will start with the general fluid flow background in the next section.

2.3 FLUID FLOW BACKGROUND

2.3.1 Single-class Traffic Fluid Flow Model

To simplify the complexity in deriving the fluid flow model, we consider a single server first-come-first-serve (FCFS) queueing system with nonstationary arrival process; μ denoting the average queue service rate and λ representing the ensemble average arrival rate at time t. The model is developed by focusing on the dynamics of the packet queues at the transmission link. With $x(t)$ defined as the state variable representing the ensemble average number in the system at time t, $\dot{x}(t) = dx/dt$ is the rate of change of the state variable with respect to time. According to the flow conservation principle, the rate of change of the average number in the system equals to the difference between the flow in and the flow out of the system at time t, denoted by $f_{in}(t)$ and $f_{out}(t)$:

$$\dot{x}(t) = -f_{out}(t) + f_{in}(t) \tag{2.2}$$

The flow in simply equals to the arrival rate $f_{in}(t) = \lambda(t)$ for infinite waiting space queue with no packets being dropped. The flow out can be related to the ensemble average utilization of

the server as $f_{out}(t) = C\rho(t)$ where C represents link capacity. The equation (2.2), sometimes referred to as *fluid flow equation*, is quite general and can model a wide range of queueing systems as shown in [14], [11]. The fluid flow equation can then be written in terms of the average arrival rate and the departure rate as:

$$\dot{x}(t) = -C\rho(t) + \lambda(t) \tag{2.3}$$

The average server utilization $\rho(t)$ in Equation (2.3) depends on the stochastic modeling assumptions of the queue under study, such as the traffic arrival process and the packet length distribution. However, in general, the exact expression of $\rho(t)$ is difficult to determine. If experimental data is available, the function may be empirically formulated by curve fitting from measurements. However, in most cases, such data is not obtainable and the function has to be determined by an approximation approach, such as matching the equilibrium point in the differential equation at particular time instant from Equation (2.3) to the corresponding steady state queueing theory result. To adopt this pointwise mapping approach, we assume that the utilization function can be approximated by the non-negative function $G(x(t))$, which represents the ensemble average utilization of the link at time t in the form of the state variable. Equation (2.3) can then be written as:

$$\dot{x}(t) = -C(G(x(t))) + \lambda(t) \tag{2.4}$$

In general, the resulting function $G(x(t))$ is non-linear and we can solve for $x(t)$ in Equation (2.4) using any standard integration routines in an iterative fashion. The idea is that given an initial condition of the state variable at time zero as $x(0)$, we divide the whole simulation time into small intervals and only solve step by step for the solution at the end of each small interval, which then becomes an initial condition for the next time step. The nonstationary condition such as the arrival rate for the new time step can be adjusted if necessary, while the procedure is repeated for each time interval along the time horizon. The accuracy of the solution depends on the step size. The smaller the step size, the higher level of accuracy the solution will be. However, it will also come with the higher computational cost in solving such equations. Numerical studies in [12] have shown that results provided from integrating

differential equations using the fourth or the fifth order Runge-Kutta method are reasonably accurate for the case considered.

2.3.2 Multiclass Traffic Fluid Flow Model

The fluid flow model in Equation (2.4) only shows the general form of the *state model* that represent the average number in the system at a queue by a single nonlinear differential equation. Until now, only a single traffic class has been considered. To apply the fluid flow modeling approach to queueing networks, we consider the case of S different traffic classes; each arriving at the node with the average arrival rate of $\lambda_1(t), \lambda_2(t), \ldots, \lambda_S(t)$ respectively as illustrated in Figure 2.2.

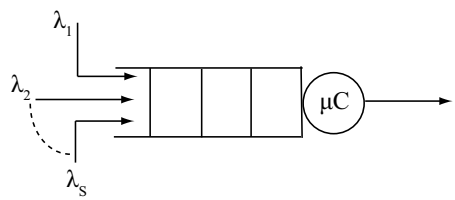

Figure 2.2: Node queuing model with S traffic classes

The aggregate traffic of certain arrival streams can be considered as one arrival process or $\lambda_T(t) = \sum_{l=1}^{S} \lambda_l(t)$. Therefore each network link can be modeled as a multi-traffic class node queue. With $x_i(t)$ represents the ensemble average number of class i in the system at time t, the total average number in the system is defined as $x_T(t) = \sum_{l=1}^{S} x_l(t)$, and the state model in Equation (2.4) becomes:

$$\dot{x}_T(t) = -C(G(x_T(t))) + \lambda_T(t) \tag{2.5}$$

To develop the fluid flow equation, we note that the flow conservation principle still applies to each traffic class; therefore, a state model in Equation (2.5) can also be developed for each class with $G(x_l(t), x_T(t))$ as the average utilization function of the link by class l traffic

in the form of the total average number in the system and the average number of class l packets in the system.

$$\dot{x}_l(t) = -C(G_l(x_l(t), x_T(t))) + \lambda_l(t) \qquad \forall l = 1, 2, \ldots, S \tag{2.6}$$

This utilization function $G_l(x_l(t), x_T(t))$ for multiple traffic classes in Equation (2.6) has to be carefully determined since it represents the portion of link capacity seen by the class l packets, compared to the total amount of link capacity being used. In the case that other traffic classes of packets are also being routed through this link, they will use part of the transmission capacity. On the other hand, if there is solely class l traffic present in the link, the average utilization function will simply be a function of the class l packets. Thus, Equation (2.6) is the general fluid flow model that describes the queue length dynamics of class l traffic in the queueing network with multiple traffic classes being present. Thus the transmission link can be described by a set of S state equations, each representing the traffic behavior of its own class. With this set of nonlinear equations describing the time varying behavior of the mean queue lengths, we can study the dynamics of the whole network.

Now considering a network consisting of M nodes, an arbitrary node i is shown in Figure 2.3. At each node, there are $M-1$ possible packet types, sorted into $M-1$ classes based on their destinations. We assume that packets are generated at the node i destined for node j according to a nonstationary arrival process, with mean rate $\gamma_i^j(t)$, and the state variable $x_i^j(t)$ as the average number of packets in the queueing system at node i destined for node j at time t. We assume a mean packet length of size μ and let C_i denote the transmission capacity of node i. When considering the network as a whole, Equation (2.6) must be modified to clearly identify the source node and destination node for each state variable, as well as to model the idea of traffic being routed through intermediate nodes when a direct link is not accessible. When considering ad-hoc networks, one often has network node serving as a relay node. A control parameter $r_{ik}^j(t)$ is thus introduced as the routing factor if traffic from node i destined to node j is routed through network node k.

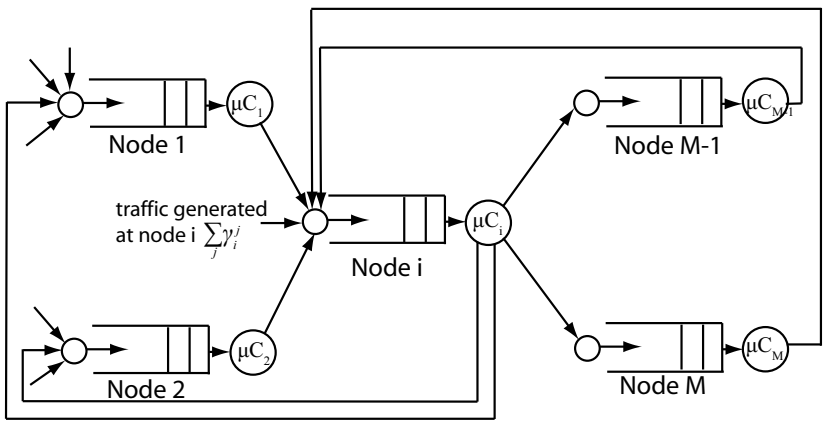

Figure 2.3: An arbitrary node i queueing model with M traffic class in an ad-hoc network

Since the flow in and flow out between node i and node k of a particular traffic class j will depend on the link between two nodes and the routing and flow controls for such traffic class, thus the state model in Equation (2.6) should be modified to incorporate the effect of link connectivity element $a_{ij}(t)$ from the adjacency matrix as well as the routing parameter $r_{ik}^j(t)$.

$$\dot{x}_i^j(t) = -C_i(G_i^j(x_i^j(t), x_T(t)))\sum_{k\neq i} a_{ik}(t)r_{ik}^j(t) + \gamma_i^j(t) + \sum_{l\neq i, l\neq j} C_l(G_l^j(x_l^j(t), x_T(t)))(a_{li}(t)r_{li}^j(t)) \quad \forall i, j = 1, 2, \ldots, M \quad (2.7)$$

In Equation (2.7), the first term to the right of the equal sign represents the flow of traffic class j out of node i, the second term denotes the type j traffic entering the network at node i, and the last term characterizes the flow of type j traffic being routed into node i from other network nodes. The two rightmost terms in Equation (2.7) represent the total class j traffic flow into the queue at node i, namely $\lambda_i^j(t)$.

2.4 SUMMARY

This chapter reviews the literature of various approaches commonly used to measure the performance of MANETs. Most of the previous works focus on steady state conditions only. There is also a concern of scalability issue as standard simulation methodology suggested that several simulation runs with different random number seeds are required to assure an accurate portrayal of the system, which leads to the requirement of large amount of computational time and resources. This chapter also includes the review of the techniques for nonstationary behavior modeling and how the approximation method can be used to estimate the time dependent performance using steady state results.

To properly study the network with dynamic behavior, we believe that the nonstationary conditions should be incorporated into the performance evaluation model. As the work in this dissertation aims to develop the performance tool based on an approximate fluid flow modeling technique, the brief background on general fluid flow is also provided.

3.0 A HYBRID APPROACH TO MODELING DYNAMIC BEHAVIOR

This chapter describes the framework of the analytical tools used in this dissertation. Since the objective of our work is to develop a new performance modeling technique that focuses on both time varying and steady state behavior, we use mobile ad-hoc networks as an example because of its characteristics of nonstationary conditions. Figure 3.1 illustrates the key components in a simulation scenario framework for mobile ad hoc networks. As shown in the figure, the main components are traffic pattern, protocol layers, queueing model, network topology, mobility model, wireless links, energy model, and performance measures. However, as the work in this dissertation mainly deals with queueing analysis of nonstationary networks, the development of the proposed performance modeling technique will be centered on the queueing model element. Other components related to the proposed model including the traffic patterns, wireless link and network topology will also be discussed accordingly.

Regarding the dynamic nature of mobile ad hoc networks, the topology can be dynamically changed depending on link connectivity between each node pair. When nodes in the networks are allowed to move arbitrarily, it will lead to frequent changes in the topology of the queueing network model. In order to study the dynamic network performance, the time varying behavior of such model must be determined. The closed form expression for the general time varying behavior of arbitrary topology queueing model is extremely difficult to derive. For that reason, we propose a new hybrid approach to approximate the network performance using two components; (1) an adjacency matrix model to represent network topology and (2) a fluid flow model based set of differential equations which model time dependent queueing behavior of each node.

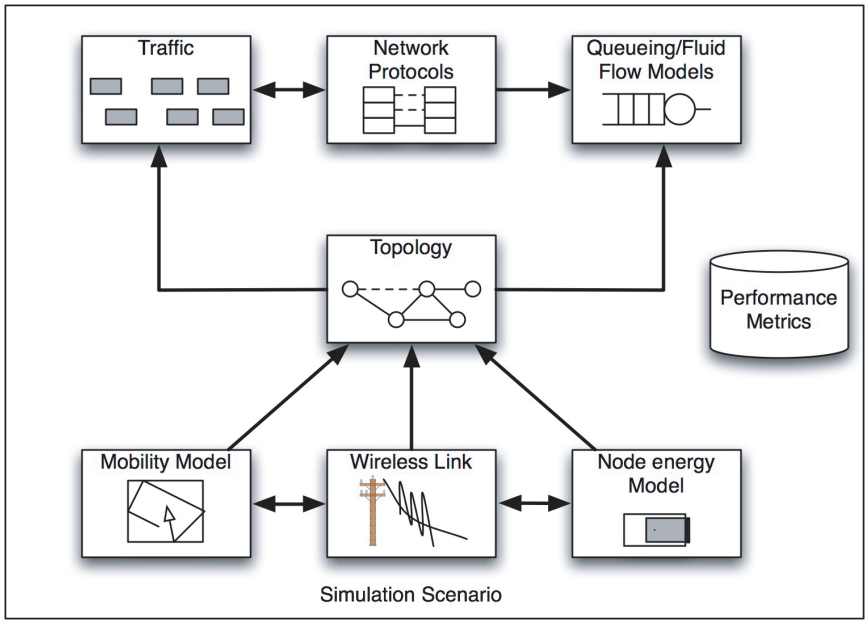

Figure 3.1: Simulation Scenario

3.1 NETWORK TOPOLOGY MODELING

Consider an ad-hoc network consists of M nodes, the network topology at any time is modeled with a M x M adjacency matrix denoted as $A(t)$.

$$A(t) = \begin{bmatrix} a_{11}(t) & a_{12}(t) & \dots & a_{1M}(t) \\ a_{21}(t) & a_{22}(t) & \dots & a_{2M}(t) \\ \vdots & \vdots & & \vdots \\ a_{M1}(t) & a_{M2}(t) & \dots & a_{MM}(t) \end{bmatrix}$$

where,

$$a_{ij}(t) = \begin{cases} 1, & \text{if node } i \text{ and } j \text{ are directly connected at time } t \\ 0, & \text{otherwise} \end{cases}$$

The connectivity between two nodes depends on their radio range which is a function of antenna patterns, interference, geographic terrain, transmission power, speed, propagation environment, receiver sensitivity, etc. The range of each node pair is generally different and the connectivity can also be asymmetric ($i.e., a_{ij}(t) = 1$ while $a_{ji} = 0$).

Most of the simulation and analytical schemes model the radio propagation of MANETs using the simplifying assumptions that all radios have a perfect coverage on a two-dimensional circular range, even with the awareness of the necessity and complexity of realistic radio propagation models. The problem of connectivity is also simplified here by assuming the connectivity to be a function of radio range R and the calculated distance d_{ij} between node i and node j;

$$a_{ij}(t) = \begin{cases} 1, & d_{ij} \leq R \text{ at time t} \\ 0, & \text{otherwise} \end{cases}$$

Taking into account the issue of node mobility, it is likely that the network will go through a series of events such as links being broken and removed from the topology as two nodes move away from each other, or links are being added to the topology when two nodes come into radio range. The dynamic network topology is reflected by changes with time in the value of the $a_{ij}(t)$ elements in the adjacency matrix. Given an initial placement of network nodes and mobility characteristics of the nodes, every node pair is checked for the possible connectivity change based on their current speeds and directions. As time progresses, every event including the changes in speed and direction are placed in chronological order, and the pair wise radio connectivity is re-calculated, which then changes the adjacency matrix accordingly. The connectivity in adjacency matrix is checked and updated periodically to reflect the nonstationary conditions through topology changes in each time period. The $a_{ij}(t)$ elements are later used to determine routing control parameters in order to forward the data packets towards their destination with the chosen routing scheme. Since only the link connectivity between each pair of nodes is needed, the concept of adjacency matrix is flexible to be used with any mobility pattern.

Information about node movement can be received either from mobility models or trace data. However, with trace data not always accessible and mobility models are computationally expensive in simulations, [36] and [37] proposed an alternative idea of using simple link connectivity to implicitly represent mobility and physical layer connectivity as well as provide a more direct relation with the network environment. Since the random waypoint model is considered one of the most commonly used mobility models for simulations of MANETs, it has been chosen to be used as an example to validate the performance of link connectivity model in comparison to traditional mobility models.

Properties of the random waypoint model have been studied many times in [38]- [39]; however, the relationship between the parameters of a mobility model and the changes in link connectivity has never been quantified which makes it difficult to design simulation experiments that control link connectivity properties using mobility model parameters. Moreover, the fact that node movement leading to a sequence of on-off link changes and such changes in link connectivity directly affecting MANET performance makes the idea of link connectivity model seems sensible. Therefore, rather than dealing with random variables that describe speed and direction for various types of mobility characteristics, each link connection for all pairs of nodes can be calculated more efficiently from a probabilistic model, specifically a two-state Markov chain with the UP state defined as the duration of links being connected and the $DOWN$ state as the time when link being disconnected. For each link connection, the link up rate is defined as the number of times the link goes from being disconnected to being connected, and in a similar way, the link down rate is the transition rate from being in the connected state to the disconnected state [36], [37].

The probabilistic link connectivity model was proposed with original intention to solve the shortcomings of long warm up period and high computational requirement needed in any standard random mobility model to compute node position and determine link connection between each node pair. The link connectivity model has fewer parameters, thus simplifying the design of simulation experiments. To be specific, only the average link lifetime parameters, namely T_{UP} and T_{DOWN}, are sufficient to fully characterize the model. Moreover,

the idea of using only these two variables seems to provide a better control in link stability characteristics than typical mobility models. The state of each link can be aggregated in an adjacency matrix to represent the topology of the network. The elements in the matrix are two-state random variables that specify the status of the link which can either be symmetric bidirectional links or asymmetric ones. Though the link connectivity model based on a two-state Markov chain addresses the computational weakness of the random waypoint model, it comes with the drawback that it does not support high-fidelity models of mobility and physical layer characteristics. However, it is believed that the practical effect of computational time reduction outweigh the possibility of potential loss of fidelity when compared to actual node mobility models.

In contrast to the idea of link connection in adjacency matrix, discrete event computer-based simulators do not retain the concept of link connectivity, but rather use routing protocols to determine routing control parameters and maintain the routing table. If the signaling packets are successfully sent from source node s to destination node d, then these two nodes can communicate to each other, which implies that $a_{sd}(t) = 1$. When node s cannot communicate with node d directly and multi-hop routing is enabled, then routing protocol will figure out a way to forward traffic through intermediate nodes to destination nodes.

3.2 NONSTATIONARY QUEUEING MODELING

The second component of the hybrid approach is to use a fluid flow based set of differential equations to model time dependent queueing behavior of each node. An overview on general fluid flow modeling has already been discussed in the previous chapter. We have seen that the average server utilization of fluid model in Equation (2.3)-(2.7) depends on stochastic modeling assumptions of the queue under study. Therefore, various queue types with different traffic arrival process and different packet length distribution will lead to separate fluid models.

3.2.1 Nonstationary Queueing Network Modeling with Exponential Packet Size

Although the fluid flow expression for single node queue and for nonstationary queueing network have been derived in Equation (2.4) and Equation (2.7) respectively, the utilization function $G(.)$ for different queue types has not been identified yet. The most commonly used stochastic queueing model assumes that the arrival process and service process are both memoryless, therefore the traffic is assumed to arrive to the queue according to a nonstationary Poisson process while the packet length is assumed to be exponentially distributed. With one traffic class, $\lambda(t)$ denotes the time varying arrival rate, $1/\mu$ refers to average packet length, and C defines the server capacity respectively. Then μC represents the average service rate in number of packets being served per unit time. Packet transmission time is also assumed to be proportional to the packet length. Hence, the system is modeled as a $M/M/1$ queue. When the arrival rate to the queue is constant, the average number in the system at steady state is given by $x(t) = \lambda/(\mu C - \lambda)$. From Equation (2.4), at steady state, when $\dot{x}(t) = 0$, which then results in $\lambda = CG(x(t))$. If substitute this back into the steady state equation and solve for the root of the equation, we get $G(x(t)) = \mu[x(t)/(1 + x(t))]$. Hence, the fluid flow equation for the one node queue with single class of traffic as given by the state model equation in (2.4) now becomes:

$$\dot{x}(t) = -\mu C\big(\frac{x(t)}{1 + x(t)}\big) + \lambda(t) \tag{3.1}$$

To generalize Equation (3.1) for multiple classes of traffic, we follow the approach from [2], [12]. Since Poisson arrival streams can be aggregated into one arrival process, leading to $\lambda_T(t) = \sum_{l=1}^{S} \lambda_l(t)$; this implies that each network link with various incoming traffic flows can be modeled as a multi-traffic class node queue. With $x_i(t)$ represents the ensemble average number of class i in the system at time t, the total average number in the system is defined as $x_T(t) = \sum_{l=1}^{S} x_l(t)$, and the state model in Equation (3.1) becomes:

$$\dot{x}_T(t) = -\mu C\Big(\frac{x_T(t)}{1 + x_T(t)}\Big) + (\lambda_1(t) + \lambda_2(t) + \ldots + \lambda_S(t)) \tag{3.2}$$

Following the same standard queueing formula at steady state, we have $x_T(t) = \lambda_T/(\mu C - \lambda_T)$ and $x_l(t)$ - the state variable representing the average number in the system of $l-th$ traffic class as:

$$x_l(t) = \frac{\lambda_l}{\mu C - \sum_{j=1}^{S} \lambda_j} \qquad \forall l = 1, 2, \ldots, S \tag{3.3}$$

To determine $G(x_l(t), x_T(t))$, we start with equation (3.2) at steady state, the derivative of the state variable $\dot{x}_T(t)$ equals to zero, which then results in $\mu C(x_T(t)/(1 + x_T(t))) = \sum_{j=1}^{S} \lambda_j(t)$. Similarly, Equation (2.6) at steady state, $\dot{x}_l(t) = 0$, yields $C(G_l(x_l(t), x_T(t))) = \lambda_l(t) \quad \forall l = 1, 2, \ldots, S$. When solving along with Equation (3.3), the average utilization function $G_l(x_l(t), x_T(t))$ can be obtained as:

$$G(x_l(t), x_T(t)) = \mu\Big(\frac{x_l(t)}{1 + x_T(t)}\Big) \tag{3.4}$$

Finally, as the flow conservation principle still applies for each traffic class, we substitute Equation (3.4) back into Equation (2.6) we get:

$$\dot{x}_l(t) = -\mu C\Big(\frac{x_l(t)}{1 + x_T(t)}\Big) + \lambda_l(t) \qquad \forall l = 1, 2, \ldots, S \tag{3.5}$$

If taking routing control parameters and link connectivity elements from adjacency matrix into account, Equation (3.5) becomes:

$$\dot{x}_i^j(t) = -\mu C_i\Big(\frac{x_i^j(t)}{1 + x_T(t)}\Big) \sum_{k \neq i} a_{ik}(t) r_{ik}^j(t) + \gamma_i^j(t) + \sum_{l \neq i, l \neq j} \mu C_l\Big(\frac{x_l^j(t)}{1 + x_T(t)}\Big)(a_{li}(t) r_{li}^j(t)) \qquad \forall i, j = 1, 2, \ldots, M \tag{3.6}$$

Comparing the fluid flow model for Poisson traffic with exponential packet size in Equation (3.6) with the general fluid flow expression in Equation (2.7), we can see that the first term to the right of the equal sign represents the flow out of node i for particular type j traffic, while the two rightmost terms represent the total flow of class j traffic into the queue at node i.

3.2.2 Nonstationary Queueing Network Modeling with Constant Packet Size

Though the assumption of traffic with Poisson arrival and exponential packet size is most commonly used when studying the behavior of queueing networks. Traffic in many scenarios may not conform to the stochastic assumption of the model. The idea of fixed length packets facilitates network design especially when dealing with congestion control and fairness issues. As mentioned earlier, the average server utilization in fluid flow Equation (2.4) depends on stochastic modeling assumptions of the queue under study such as traffic arrival process and packet length distribution. Assuming the traffic arrival still remains to be a nonstationary Poisson process, the packet length however no longer has exponential distribution but instead being fixed constant. With $\lambda(t)$ denotes the time varying arrival rate, $1/\mu$ refers to average packet length, and C defines the server capacity respectively, μC represents the constant service rate in number of packets being served per unit time. Packet transmission time is also assumed to be proportional to the packet length.

From queueing theory, the ensemble average number in the system at steady state for a $M/D/1$ queue is given by $x(t) = \rho + \rho^2/[2(1-\rho)]$, where $\rho = \lambda(t)/(\mu C)$. Following the same technique of matching the steady-state equilibrium points to obtain the utilization function as in the previous section results in the fluid flow model in Equation (2.4) becoming:

$$\dot{x}(t) = -\mu C[x(t) + 1 - \sqrt{x(t)^2 + 1}] + \lambda(t). \tag{3.7}$$

Having established the model for single traffic class of this particular queue, it is possible to generalize Equation (3.7) for multiple traffic class using the same approach as in previous section. By adding up Poisson arrival streams into one aggregate arrival process, the state model becomes:

$$\dot{x}_T(t) = -\mu C[x_T(t) + 1 - \sqrt{x_T(t)^2 + 1}] + \lambda_T(t). \tag{3.8}$$

Let $x_l(t)$, and $\lambda_l(t)$ represent the average number of class l packets in the system and arrival rate of class l packets respectively, while $x_T = \sum_{l=1}^{S} x_i$ and $\lambda_T = \sum_{l=1}^{S} \lambda_i$ denote the total average number in the system and the mean aggregate arrival rate to the queue. Note that at steady state, the ensemble average of the total traffic in the system equals to

$$x_T(t) = \frac{\lambda_T}{\mu C} + \frac{\lambda_T^2}{2\mu^2 C^2 (1 - \frac{\lambda_T}{\mu C})}. \tag{3.9}$$

We can also write $x_l(t)$ - the state variable representing the average number in the system of class l traffic at steady state as:

$$\begin{aligned} x_l(t) &= \frac{\lambda_l}{\lambda_T}[\frac{\lambda_T}{\mu C} + \frac{\lambda_T^2}{2\mu^2 C^2 (1 - \frac{\lambda_T}{\mu C})}] \\ &= \frac{\lambda_l}{\mu C} + \frac{\lambda_l \lambda_T}{2\mu^2 C^2 (1 - \frac{\lambda_T}{\mu C})} \\ &= \frac{\lambda_l [2\mu C - \lambda_T]}{2\mu C [\mu C - \lambda_T]}. \end{aligned} \tag{3.10}$$

To derive the average utilization function $G(x_l(t), x_T(t)$, start with Equation (3.8) at steady state, $\dot{x}_T(t)$ equals zero, which then results in $\lambda_T(t) = \mu C[x_T(t) + 1 - \sqrt{x_T^2 + 1}]$. Similarly, from Equation (2.7), $\dot{x}_l(t) = 0$, we get $C(G_l(x_l(t), x_T(t)) = \lambda_l(t) \quad \forall l = 1, 2, \ldots, S$. Solving these two equations along with Equation (3.10), the average utilization function for class l packets, $G_l(x_l(t), x_T(t))$ can be obtained as:

$$G_l(x_l(t), x_T(t)) = \frac{2x_l(t)[\sqrt{x_T^2(t) + 1} - x_T(t)]}{\sqrt{x_T^2(t) + 1} - (x_T(t) - 1)}. \tag{3.11}$$

Substituting the last equation back into the general fluid flow expression in Equation (2.6), we get:

$$\dot{x}_l(t) = -\mu C[\frac{2x_l(t)[\sqrt{x_T^2(t) + 1} - x_T(t)]}{\sqrt{x_T^2(t) + 1} - (x_T(t) - 1)}] + \lambda_l(t) \qquad \forall l = 1, 2, \ldots, S \tag{3.12}$$

As a result, Equation (3.12) is the set of fluid flow equations describing the queue length dynamics of each class l traffic in the queueing network consisting of multiple traffic classes when input traffic is Poisson arrival with constant packet length. When incorporating the

effect of link connectivity and routing control parameters if traffic needs to be relayed through other neighboring nodes, Equation (3.12) will become:

$$\dot{x}_i^j(t) = -\mu C\Big(\frac{2x_i^j(t)[\sqrt{x_T^2(t)+1} - x_T(t)]}{\sqrt{x_T^2(t)+1} - (x_T(t)-1)}\Big)\sum_{k\neq i} a_{ik}(t)r_{ik}^j(t) + \gamma_i^j(t) + \sum_{l\neq i, l\neq j} \mu C_l\Big(\frac{2x_l^j(t)[\sqrt{x_T^2(t)+1} - x_T(t)]}{\sqrt{x_T^2(t)+1} - (x_T(t)-1)}\Big)(a_{li}(t)r_{li}^j(t)) \qquad \forall i,j = 1,2,\ldots,M \quad (3.13)$$

An assumption of deriving Equation (3.13) is that the aggregate traffic arrival process at each node is Poisson. However, this is an approximation since the output of the deterministic service process will not be Poisson. In some cases, Equation (3.13) may still be applicable when traffic from exogenous sources at each node, i.e., $\gamma_i^j(t)$, has higher weight values than incoming relayed traffic from other neighboring nodes. However, in general, the assumption of Poisson aggregate traffic will result in fluid flow models which overestimate the average number in the system.

To solve such a problem, it was suggested in [3], [40], [41], [42], [43] that output from queueing system with deterministic service time should be treated as an delayed input to the next stage. This idea is applicable to our model when the packet size is fixed. In this case, the input to the next stage is basically nothing but a superposition of the delayed input streams from the nearby nodes plus any external arriving traffic $\gamma_i(t)$. Consider a simplified two stage tandem queue model as in Figure 3.2(a)-(b), let $x_i(t), \lambda_i(t)$ and $G_i(t)$ be the average number in the system, total average arrival rate and average utilization at node i respectively. Then $\lambda_1(t) = \gamma_1(t)$ is the arrival rate to the first queue, and $CG_1(t)$ is the departure rate from the first queue. The departure rate then becomes the input the second queue with a deterministic propagation delay of D time units, or $\lambda_2(t) = CG_1(t-D)+\gamma_2(t)$. We can then write a set of differential equations to represent the rate of change of average number at node 1 and node 2, $x_1(t)$ and $x_2(t)$ for Figure, 3.2(a)-(b) as:

$$\begin{aligned} \dot{x}_1(t) &= -CG_1(t) + \gamma_1(t) \\ \dot{x}_2(t) &= -CG_2(t) + \gamma_2(t) + CG_1(t-D) \end{aligned} \quad (3.14)$$

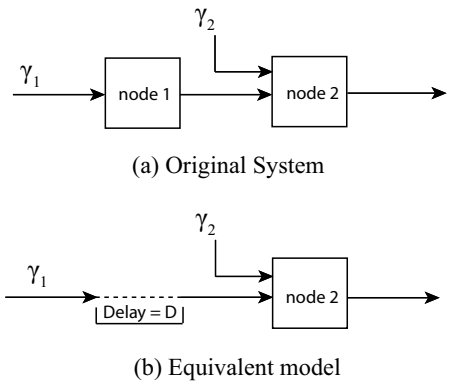

Figure 3.2: A 2-node deterministic service queue, with its equivalent model

With the knowledge of the delay adjustment for average arrival rate and departure rate at each node queue, the fluid flow model derived in Equation (3.13) can be re-written as:

$$\dot{x}_i^j(t) = -\mu C\Big(\frac{2x_i^j(t)[\sqrt{x_T^2(t)+1} - x_T(t)]}{\sqrt{x_T^2(t)+1} - (x_T(t)-1)}\Big)\sum_{k\neq i} a_{ik}(t)r_{ik}^j(t) + \gamma_i^j(t) +$$
$$\sum_{l\neq i, l\neq j} \mu C_l\Big(\frac{2x_l^j(t-D)[\sqrt{x_T^2(t-D)+1} - x_T(t-D)]}{\sqrt{x_T^2(t-D)+1} - (x_T(t-D)-1)}\Big)(a_{li}(t)r_{li}^j(t)) \qquad \forall i,j = 1,2,\ldots,M \quad (3.15)$$

Numerical results in the next chapter will show that the fluid flow model with delay D to compensate on deterministic service time can correctly estimate the number of packets in each node buffer, compared to results from standard simulators.

3.2.3 Models for Systems with the Superposition of Periodic Arrival Streams

Case I: Identical Sources The queueing models derived in previous sections using a Poisson arrival process have the advantage of simplicity and have shown to be a good approximation for system with exponentially distributed external arrivals. However, in this section, we consider another type of general queueing model in telecommunications when constant bit rate streams from different sources are segmented into fixed-length packets, then multiplexed

on a transmission link. The idea of multiplexing traffic from a large number of independent sources with different characteristic onto transmission links is comparable to the concept of single server, deterministic service time queueing systems. In the case where all the sources have the same period and the packets have the same size, the model is often referred to as the *N*D/D/1* queue.

If dealing with only periodic arrival stream from one source arriving at a single server, the system can be modeled as *D/D/1*. With the inter-arrival time between packets equal to $1/\lambda$ and the service time is $1/\mu$, number of packets in system at time t is given by

$$x(t) = [\frac{t}{1/\lambda}] - [\frac{t - \frac{1}{\lambda}}{1/\mu}] \tag{3.16}$$

where $[y]$ denotes the integer part of y. In this particular queue, assuming a stable system with $\lambda < \mu$, as each packet arrives at the server, it will either get service right away or wait for the one in the server to finish and the queue length will only fluctuate between 0 and 1. It is not easy to follow the same approach as in the previous section, solving Equation (3.16) for the close form utilization function $G(x(t))$ for this specific scenario to be used in fluid flow expression as in Equation (2.4), and later on expand the model for the multi-classes traffic. However, the utilization function can probably be found using numerical analysis on *N*D/D/1* queue length distribution.

In [44]-[45], with the buffer requirement as the considered performance measure, the exact formulae for queue length distribution when all the sources have the same period is provided, while in the case of sources with different periods, upper and lower bounds on the queue length distribution are given. Consider a single server with the capacity of one packet per time unit handling multiplexed traffic from N sources, each with the same periodic arrival process. Stationarity holds because we assume that the first arrival of each flow is independently and uniformly distributed over the first period interval. Let L_t be the number of customers present at time t and $N(u, t)$ be the number of customer arrivals in the interval

(u, t). The complementary distribution of the number in the system at any arbitrary instant t: $Q_t(r) = Pr\{L_t > r\}$ is defined as the sum:

$$Q_t(r) = \sum_{s=1}^{\infty} Pr\{N(t-s,t) = r+s \text{ and } L_{t-s} = 0\} \tag{3.17}$$

Using the binomial distribution as the probability of exactly $r+s$ arrivals in an interval of length s and the condition of queue being idle at the beginning of the interval, $Q_t(r)$ can be re-written as:

$$Q_t(r) = \sum_{s=1}^{N-r} \binom{N}{r+s} \left(\frac{s}{D}\right)^{r+s} \left(1 - \frac{s}{D}\right)^{N-r-s} \left(\frac{D-N+r}{D-s}\right) \quad \text{for } 0 \leq r < N \tag{3.18}$$

where D represents the ratio of the multiplex capacity to the arrival stream bit rate, or equivalently $D = \mu/\lambda$. To take advantage of complimentary distribution function already in hand, the average number of packets can alternatively be expressed using *Survival Function* or $Q_t(r)$ instead of the traditional probability density [43]. Recall that (see proof in Appendix)

$$x(t) = \sum_{r=0}^{N-1} Q_t(r) \tag{3.19}$$

Since the complementary distribution $Q_t(r)$ in Equation (3.18) already accounts for the case of N multiplex traffic, using Equation (3.19) would result in the total average number in the system $x_T(t)$. Solving Equation (3.18) and (3.19) together, we are able to determine the system traffic load and its corresponding total average number in the system for any given N traffic classes. Knowing the information pair (λ_T, x_T) at each steady state equilibrium point, we then apply the data fitting approach using these numbers to find the utilization function $G(.)$ which is supposed to be in the terms of $x(t)$ to express the effect of different load on the system's servers. The resulting $G(x_T(t))$ is in the form of a nonlinear polynomial expression (i.e., $G(x_T(t)) = ax_T^n + bx_T^{n-1} + ... + k$) and can be plugged back into general fluid flow expression in Equation (2.5). The utilization function from data fitting process is not in the general close form that would be applicable to any $N * D/D/1$ scenario as it depends on the specific λ and μ of periodic arrival streams modeled. Note that the process can be

done offline.

Since all the sources are identical, we can also write $x_l(t)$ - the state variable representing the average number in the system of class l traffic at the steady state condition as $x_l(t) = (\lambda_l/\lambda_T)x_T(t)$. Following the same approach of steady state equilibrium matching, from (2.5), $\dot{x}_T(t) = 0$, we will get $\lambda_T(t) = CG(x_T(t))$ and from (2.7), $\dot{x}_l(t) = 0$, we will get $CG_l(x_l(t), x_T(t)) = \lambda_l(t) \quad \forall l = 1, 2, \ldots, S$. Solving these equations along with the resulting $G(x_T(t))$ already received from the data fitting, the average utilization for class l packets, $G_l(x_l(t), x_T(t))$ can be written as:

$$\begin{aligned} G(x_l(t), x_T(t)) &= \frac{x_l}{x_T} * G(x_T(t)) \\ &= \frac{x_l}{x_T}[ax_T^n + bx_T^{n-1} + ... + k] \end{aligned} \tag{3.20}$$

If substitute the last equation back into the fluid flow model in Equation (2.6), we get:

$$\dot{x}_l(t) = -\mu C \frac{x_l}{x_T}[ax_T^n + bx_T^{n-1} + ... + k] + \lambda_l(t) \quad \forall l = 1, 2, \ldots, S \tag{3.21}$$

Equation (3.21) represents the set of fluid flow equations describing the queue length dynamics of each class l traffic in the queueing network consisting of multiple periodic arrival streams. After incorporating the effect of routing parameters, link connectivity element from adjacency matrix, as well as delay adjustment required for queueing system with deterministic service time, the final result for fluid expression can be described as:

$$\begin{aligned} \dot{x}_i^j(t) = -\mu C \frac{x_i^j(t)}{x_T(t)}[ax_T^n(t) + bx_T^{n-1}(t) + ... + k] \sum_{k \neq i} a_{ik}(t) r_{ik}^j(t) + \gamma_i^j(t) + \\ \sum_{l \neq i, l \neq j} \mu C_l \frac{x_l^j(t-D)}{x_T(t-D)}[ax_T^n(t-D) + bx_T^{n-1}(t-D) + ... + k](a_{li}(t) r_{li}^j(t)) \quad \forall i, j = 1, 2, \ldots, M \end{aligned} \tag{3.22}$$

Case II: Non-identical Sources Now we consider the case where a group of sources with different rates are multiplexed on a transmission link where the total bit rate is less than the transmission capacity to ensure stability. Suppose there are m types of sources with N_i of type i generating traffic at the rate of one packet per D_i time units, for $i = 1, 2, ..., m$. This type of queue may be referred to as $N_1D_1 + N_2D_2 + ... + N_mD_m/D/1$ queue. Similar to

the case of periodic arrival streams from identical sources, the fluid flow model to describe the dynamics of the average number in the system for the non-identical sources is difficult to derive and we still need to rely on numerical analysis of the queue length distribution to find the utilization function. Nevertheless, when arrival traffic comes from various sources with different periods, the exact formula for the queue length distribution as in the case of identical sources cannot be obtained and the upper and the lower bounds on such distribution will be given instead [44]-[46].

Following the same idea of Equation (3.17) to derive the complementary system occupancy distribution at an arbitrary instant t, we need to find two probability components; $p_s(r) = Pr\{N(t-s,t) = r+s\}$ and $\pi_0(r,s) = Pr\{\text{system empty at } t-s|\ r+s \text{ arrivals in } (t-s,t)\}$. Equation (3.17) can be written in a more compact form as:

$$Q_t(r) = \sum_{s=1}^{\infty} p_s(r)\pi_0(r,s) \tag{3.23}$$

For an interval of length s, a type i stream will generate $[s/D_i]$ packets for certain, and probably one additional packet with probability $\alpha_i = s/D_i - [s/D_i]$, where $[y]$ represents the integer part of y. Considering all N_i of type i sources, the total number of arrivals equals to $N_i[s/D_i] + k_i$ where k_i is a random variable representing additional packets for each traffic type with the following distribution:

$$b_{si}(k) \quad = \quad Pr\{k_i = k\} = \begin{cases} \binom{N_i}{k}\alpha_i^k(1-\alpha_i)^{N_i-k} & \text{for } 0 \le k \le N_i \\ 0 & \text{otherwise} \end{cases} \tag{3.24}$$

Then $q_s(k)$ is defined as the distribution of $\sum k_i$ for all m types.

$$q_s(k) = \sum_{\sum_i k_i = k} \prod_{i=1}^{m} b_{si}(k_i) \qquad \text{for } 0 \le k \le \sum N_i \tag{3.25}$$

Similar to the case of sources having the same period, we are looking for the complementary distribution of the number in the system at instant t: $Q_t(r) = Pr\{L_t > r\}$ as defined in

equation (3.17) and (3.23). However, in this situation, the probability of $r+s$ arrivals during $(t-s,t)$ time interval can be studied through the behavior of $q_s(k)$ instead.

$$p_s(k) = \begin{cases} q_s(r+s-\sum_i N_i[s/D_i]) & \text{for } \sum_i N_i[s/D_i] \leq r+s \leq \sum_i N_i([s/D_i]+1) \\ 0 & \text{otherwise} \end{cases} \tag{3.26}$$

The conditional probability of queue being idle at the beginning of the interval or π_0 in part of equation (3.23), is difficult to derive; however, the bounds for probabilities $Q_t(r)$ can be obtained from the fact that $(1-\rho) \leq \pi_0 \leq 1$ where ρ corresponds to the server utilization $\sum N_i/D_i$. Substituting in (3.23), we have the bounds:

$$(1-\rho)\sum_{s=1}^{\infty} p_s(r) \leq Q_t(r) \leq \sum_{s=1}^{\infty} p_s(r) \tag{3.27}$$

These inequalities in (3.27) provide queue length distribution to within an order of magnitude, which is sufficient for buffer allocation to an overflow probability of less than 10^{-10} [44]. The tighter upper bound has been rigorously derived in [44]-[46] as follow:

$$Q_t(r) \leq \sum_{s=1}^{\infty} \Big\{ \sum_{\sum k_i = r+s-d_s} \prod_{i=1}^{m} b_{si}(k_i) \Big(1 - \sum_{i=1}^{m} \frac{N_i - k_i}{D_i(1-\alpha_{si})}\Big)^+ \Big\} \tag{3.28}$$

with $(x)^+ = \max\{0,x\}$ and $d_s = \sum N_i[s/D_i]$. Since we assume that the first arrival of each flow is independently and uniformly distributed over the first period interval, stationarity yet again holds here. Therefore using inequalities in Equation (3.27) or (3.28) coupled with Equation (3.19), we can find the corresponding bounds for the total average number in the system. Though the expression for a tighter upper bound was suggested in (3.28), the lower bound inequality for queue length distribution in Equation (3.27) does not make use of any information of the k_i combination to tighten the bound. For the system with a high server utilization, the inequalities in Equation (3.27) result in a wide gap between bounds. Another way to possibly reduce the gap between bounds is by assuming homogeneous traffic. For the lower bound, traffic from all sources are fixed with period of D_{max} and for the upper bound, the traffic period equals to D_{min} where $D_{max} = \max\{D_i\}$ and $D_{min} = \min\{D_i\}$.

Similar to the case in previous section, fitting the data pair (λ_T, x_T) at each steady point will result in polynomial utilization function $G(x_T(t))$, which can be substituted back to the

fluid flow model in Equation (2.5). However, such a utilization function is not in the general close form that is appropriate to any $\sum_i^m N_i D_i/D/1$ queueing system, the function indeed depends on λ_i and μ_i of various periodic input streams. The final result for the fluid flow expression for the nonstationary queueing system with non-identical periodic arrival streams will be in a form similar to (3.22).

3.3 SOLUTION OF THE MODEL

With the key elements in the hybrid performance model already identified, namely the network topology and the utilization function in the nonstationary queueing model, Figure 3.3 depicts the flowchart of how all the pieces can work together to solve for the time dependent performance metric of interest. We start by identifying an initial condition for the state variable at time t_0 as $x(t_0)$ and the initial system capacity as $C(t_0)$. Using the pointwise stationary approximation method, any nonstationary factor such as the arrival rate is estimated to be constant over a small time interval $\Delta t, \lambda(t) = \lambda(t_0 + \Delta t/2)$. The information on the network topology from adjacency matrix together with the routing algorithm is used to determine how the traffic should be routed toward the destination, (i.e., $r_{ik}^j(t)$). Then we apply the Runge-Kutta algorithm to solve for the state variable at the end of the time interval $x(t + \Delta t)$, which then becomes the initial condition for the next time step $[t_0 + \Delta t, t_0 + 2\Delta t]$. If necessary, the link capacity, the traffic arrival rate, network topology elements and routing control parameters will be adjusted for the new time step. Then the solution to the fluid model is calculated based on the updated parameters and the procedure will be repeated until the end of the simulation process. Please note that any standard numerical integration method can be used to solve such mathematical problem numerically; however, the Runge-Kutta algorithm is one of the widely used methods and the results in Chapter 5 show that outcomes using the fourth or the fifth order are reasonably accurate for the cases considered.

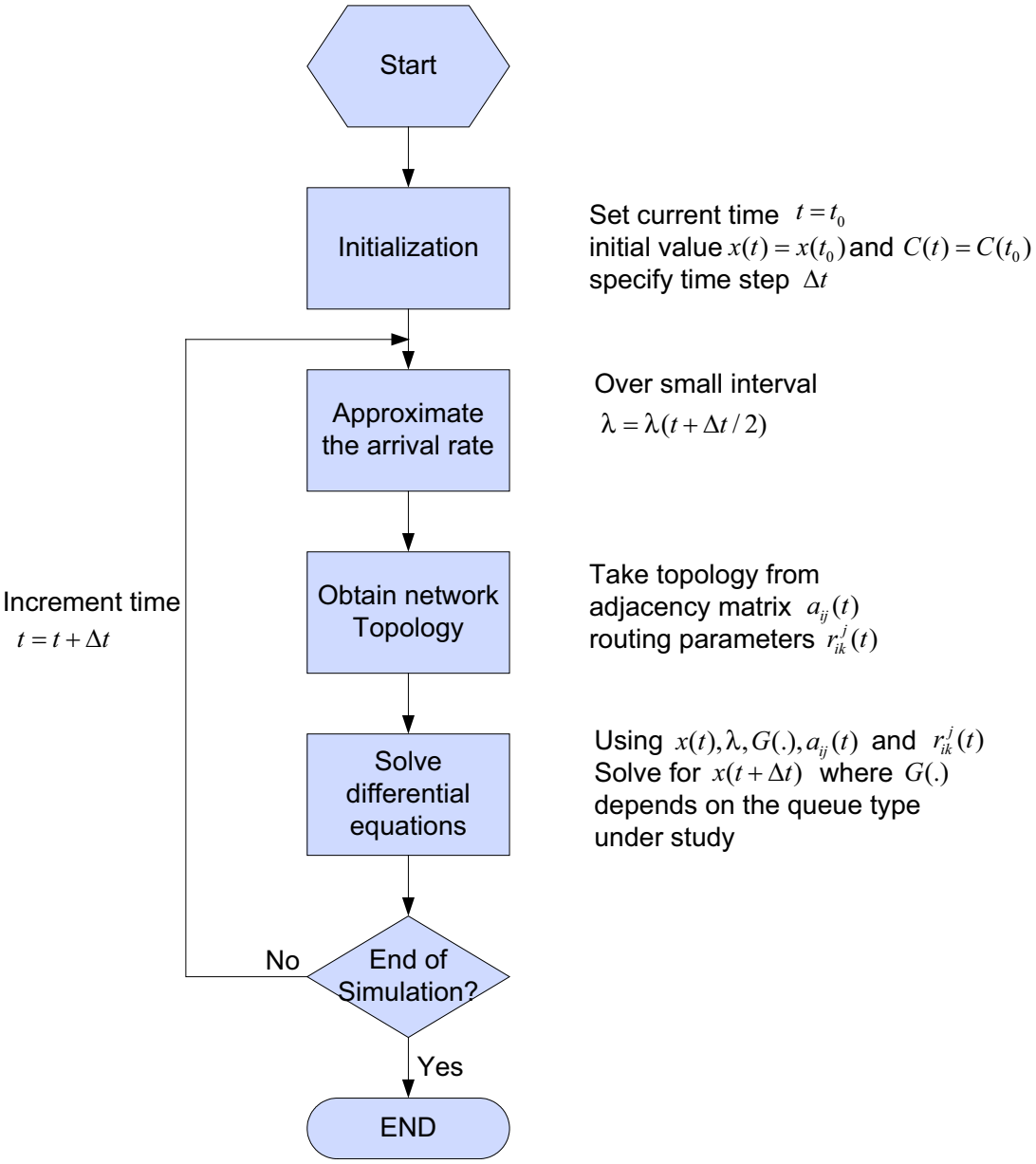

Figure 3.3: Flow chart for solving hybrid fluid-based model over $[t_0, t_f]$

4.0 OTHER PERFORMANCE MEASURES USING HYBRID APPROACH

In this chapter, we show that one can use the proposed fluid flow approach to estimate other metrics of interest, specifically end-to-end delay and the packet delivery rate. The fluid flow based approximation method is a flexible analytical tool that is not limited to only the average number in the system but can be extended to analyze the effect of other QoS metrics.

4.1 END-TO-END DELAY

In addition to the mean behavior of the network queues, delay is a commonly used QoS metric of interest in network performance. Typically, a packet needs to be forwarded from the source via a path which may include several intermediate nodes until it reaches the destination. As a result, the end-to-end delay is the sum of delays experienced at each hop along the way. The delay at a node consists of the processing time and the queueing delay at the node server, the transmission time on the link and the propagation time over the link to the node. In general, the processing delay and propagation delay are very small while the queueing delay is considered the main factor of the node's delay and depends on the state of the queue. From Little's theorem, the average number in the system is equivalent to the product of the arrival rate and the time a packet spends in the system on average. If L denotes the average number in the system, λ is the average arrival rate and W is the average waiting time, then $L = \lambda W$. With the assumption of constant arrival rate over a small step, the change in average waiting time can be related to the rate of change in average number in the system $\dot{W} = \dot{x}/\lambda$ when $\dot{x}(t)$ already defined earlier. The waiting time at each node,

referred to as *node delay*, generally includes the queueing delay for packets in the buffer and the service time of the packet currently at the node server. Since $\dot{x}(t)$ requires various average server utilization functions depending on stochastic modeling assumptions of the queue under study, this confirms our idea that the average node delay is a variable part in end-to-end delay that depends on the characteristics of the queue. The propagation delay on the other hand is considered a less significant part and often is left out of the end-to-end delay equation for simplicity.

Consider a path P of k hops from source $s = n_1$ to destination $d = n_k$, given by $(n_1, n_2), (n_2, n_3), ..., (n_{k-1}, n_k)$, where (n_i, n_{i+1}) represents a link on the path, for all $i = 1, 2, ..., k$. The average node delay at node n_i for link (n_i, n_{i+1}) is denoted by $W_i^{i+1}(t)$ and the total latency of path P if only considering average node delay can be written as $W_P(t) = \sum_{i=1}^{k-1} W_i^{i+1}(t)$. However, as we are more interested in the time dependent behavior, the rate of change of this path delay is given by:

$$\dot{W}_P(t) = \sum_{i=1}^{k-1} \dot{W}_i^{i+1}(t) = \sum_{i=1}^{k-1} \frac{\dot{x}_i^{i+1}(t)}{\lambda_i^{i+1}(t)}. \tag{4.1}$$

In general, the propagation delay is considered fixed and is almost equal for each hop on the path. If necessary, it could be added to the total node delay from Equation (4.1), as a result the end-to-end delay of path P with k hops where $T_{i,j}$ represents the propagation link delay of link (i, j) can be written as:

$$D_P(t) = W_P(t) + k * T_{i,j}(t). \tag{4.2}$$

As mentioned earlier, the average node delay part in the end-to-end delay in Equation (4.1) is related to the average number in the system in Equation (2.7), which in turn depends on the characteristics of the queue under study.

4.2 PACKET DELIVERY RATIO

Most of the work in developing the hybrid performance modeling technique in the previous chapter was on deriving the various utilization function to be used with different types of nonstationary queues. While the change in network topology is allowed and represented by elements in adjacency matrix, only bidirectional radio communication is assumed so far. It means that if node i and node j are in the radio range of each other, they can hear each other perfectly without any loss. However, an actual radio communication is not always symmetric and may exhibit diverse link quality in terms of error rate or latency. To represent a more realistic link quality, the hybrid performance model can easily incorporate the link characteristic into the elements of the adjacency matrix. The adjacency matrix does not necessarily remain as the binary matrix, which means that the connectivity a_{ij} will no longer be either 1 if nodes are directly connected or 0 otherwise. In fact, a_{ij} can be any real number between 0 and 1 to indicate the effect of link error.

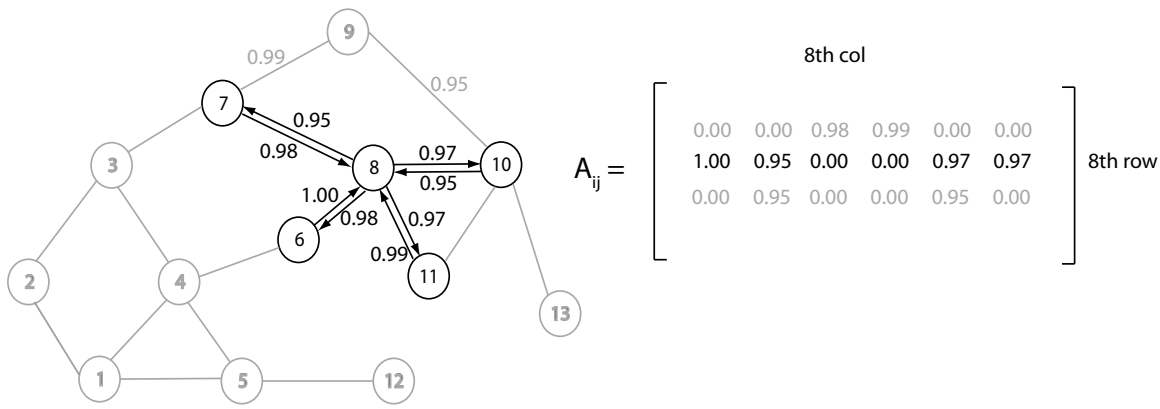

Figure 4.1: Example of using adjacency matrix to represent network with link error

With the link-level errors being included to the model, packets may be corrupted and dropped. The hybrid fluid-based models derived in the previous chapter in Equation (3.6), (3.15) and (3.22) are still valid and can still be used to approximate the time dependent average number with the link error already reflected in the link connectivity element $a_{ij}(t)$ of the adjacency matrix. However, please note that as the probability of error increases, the

more that arrival traffic will deviate from its assumed stochastic pattern.

In additional to the average number in the system, the packet delivery ratio is also an appropriate choice to evaluate the performance of the network with loss. The packet delivery ratio (PDR) is an overall performance metric which is defined as the ratio between the number of packets received by the destination nodes divided by the number of packets transmitted by the source nodes. In general, any standard simulator will return PDR as one single number by counting over the length of simulation time to obtain the total number of packets sent out from the sources and the total number of packets received at the destinations. Typically, it is present as a steady-state value.

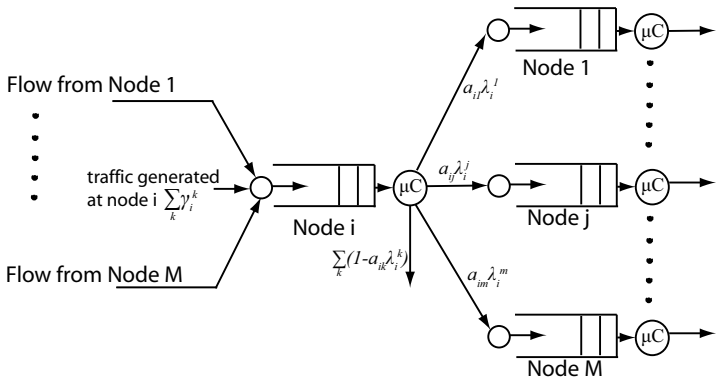

Figure 4.2: An arbitrary node i queueing model with link loss probability

From Figure 4.2, let $(1 - a_{ij})$ be the link error probability between node i and node j when $a_{ij}(t)$ is the same as the connectivity element from adjacency matrix, therefore, $\sum_k (1 - a_{ik}(t))\lambda_i^k(t)$ represents the total packets being dropped at node i due to the link error probability from all possible outgoing links. Consider a path P of k hops from source $s = n_1$ to destination $d = n_k$, given by $(n_1, n_2), (n_2, n_3), ..., (n_{k-1}, n_k)$, where (n_i, n_{i+1}) represents a link on the path, for $i = 1, 2, ..., k$, and the link error probability between node n_i and node n_{i+1} is denoted by $(1-a_{i,i+1}(t))$. The packets from node n_i will be received at the next relay node n_{i+1} with probability $a_{i,i+1}(t)$. While the number of packets being sent out

can be described by the arrival rate $\gamma_1^k(t)$, the number of packets received at the destination is related to $\prod_{i=1}^{k} a_{i,i+1}\gamma_1^k(t) * r_{i,i+1}^k$. When considering traffic from every source-destination node, the packet delivery ratio can be calculated from the following equation:

$$PDR = \frac{\sum_{i,j}(\prod_k a_{i,k} * \gamma_i^j * r_{i,k}^j) * T}{\sum_{i,j} \gamma_i^j * T} \qquad \forall i, j \tag{4.3}$$

where i is the source node, j is the destination node, k represents each relaying node on the path from source to destination and T denotes the total simulation time. Instead of single number of the packet delivery ratio usually returned by any standard simulator as shown in Equation (4.3), an alternative approach is to employ the concept of data flows. By adopting pointwise mapping approach with an assumption of constant arrival rate over a small time interval t, it is possible to describe the packet delivery ratio as a time dependent function of link error effect on the network performance by integrating the rate of packets received at the destination divided by the integration of the arrival rate from the starting point of the simulation to that particular time instant t subtracted by any packets still remain in the system due to the propagation delay δ. One advantage of using the packet delivery ratio as the performance measure is that it is a general metric which can be used with any communication networks regardless of the type of queue being employed.

$$PDR_{i,j}(t) = \frac{\int_0^t \prod_k a_{i,k}(t) * \gamma_i^j(t) * r_{i,k}^j(t)dt}{\int_0^t \gamma_i^j(t-\delta)dt} \qquad \forall i, j \tag{4.4}$$

To verify our assumption that the packet delivery ratio can be used as a performance measure to describe the time dependent behavior of the network in addition to as an overall performance metric of the whole system. We conducted a simple experiment of two-node network with probability of the link error set as 10%. For each simulation run, the standard simulator will only collect the total number of packets sent by the source node and the total number of packets being received at the destination to calculate the packet delivery ratio. Then with multiple runs using different random number seeds, the collected data is averaged over the runs to obtain the result. For example, with 1000 runs, the simulator returns the statistical summary as follows: at the end of the simulation, the total of 51.004

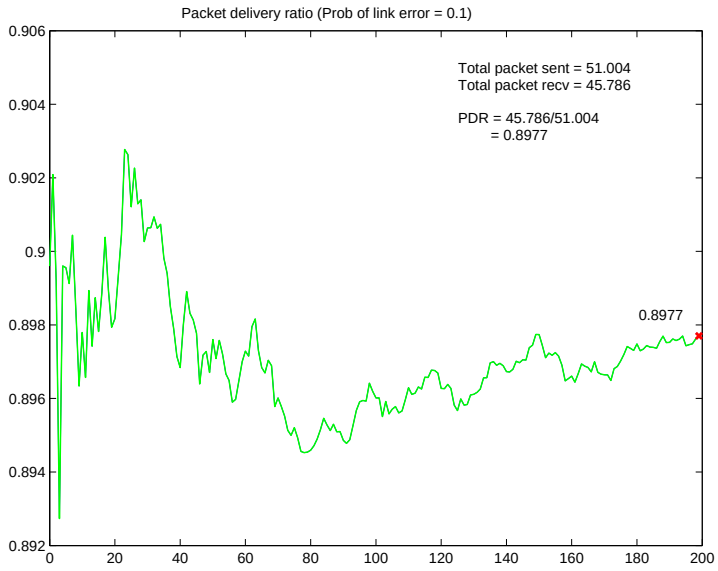

Figure 4.3: Packet delivery ratio on simple two-node network

packets were sent out, the total of 45.786 packets were received; therefore, the average packet delivery ratio equals to 0.8977. However, as we can see from Figure 4.3, the packet delivery ratio actually varies with time, depending on the relation between the cumulative number of packets being sent out to the cumulative number of packets being received from the point that the simulation starts up to that particular time instant. At the beginning, the function of packet delivery ratio is high, probably because no packet was dropped yet. As the number of packets dropped increases to reflect the link error, it will show as the decrease in total number of packets received at the destination and lower packet delivery ratio. At the end of the simulation, the result of the plot will be the same as the average number returned by a standard simulator. Therefore, we believe that it may be more appropriate to use the fluid flow method to capture its dynamic behavior, this time in terms of packet delivery ratio. The numerical results provided in the next chapter will show that the fluid flow modeling can be modified to represent other aspects of network performance including end-to-end delay and packet delivery ratio, and correctly estimate the value of such metrics compared to results from simulations.

Any packet resulted from probability of link error is flagged so that the node server can handle them properly. The current hybrid model simplifies the problem by dropping the erroneous packets without the attempt of retransmission; therefore the packets will be counted towards the total number of packet loss. The transmission rate that a node used to send a packet to the next node is fixed to be a constant. However, in a more practical model, after a failed attempt, the source node will try to transmit the erroneous packet again up to a certain number of times before giving up. In addition, the transmission rate should be adjusted based on the channel quality estimation, rather than being fixed at any particular value.

Regarding the adaptive rate mechanism, an adaptive multirate auto rate fallback protocol has been proposed for IEEE 802.11 WLANs to achieve the highest appropriate transmission rate given the condition of wireless channels. The reason is that the higher transmission rate leads to shorter transmission time and lower probability of collision, which results in less power consumption. In adaptive multirate protocol, each data rate is assigned a unique success threshold, which is used as a decision factor on when to switch to the next higher rate. The scheme to select the rate is usually based on statistical information of the received data, for example, if two consecutive Acknowledgement (ACK) frames are not received, the sending node will adapt its subsequent transmissions to the next lower rate and start the timer. On the other hand, after a certain number of successful ACKs or the timer expires, the transmission rate will be increased to the next higher one. Since the adjacency matrix components in the hybrid fluid based approach can be used to reflect the quality of wireless channel between each node pair, it is possible to incorporate the concept of adaptive link capacity into the hybrid model with the implementation of ACK and timer.

5.0 RESULTS AND DISCUSSION

The fluid flow models for nonstationary queueing systems presented in the previous section, are implemented with the use of a fifth order *Runge-Kutta* method as the standard numerical integration routine. The results of our analytical based models are validated by comparing them with results from standard simulators. We have also verified that our models and the simulation results both conform to their theoretical counterparts under stationary conditions.

5.1 EXPERIMENTAL SETUP

Sample network scenarios of three nodes, four nodes, five nodes and thirteen node topologies were studied and illustrated in Figure 5.1. We test the model on different network sizes to assess their scalability. The time varying behavior of the network in terms of the dynamics of average number in the system is studied by using Equation (2.7) with different utilization functions depending on the stochastic assumptions of the queue under study. In the previous section, we have derived the specific utilization functions for Poisson arrival traffic with exponential packet size, Poisson arrival traffic with constant packet size, periodic arrival streams when all sources share the same period and periodic arrival streams when sources may have different periods as in Equation (3.6), (3.15), and (3.22).

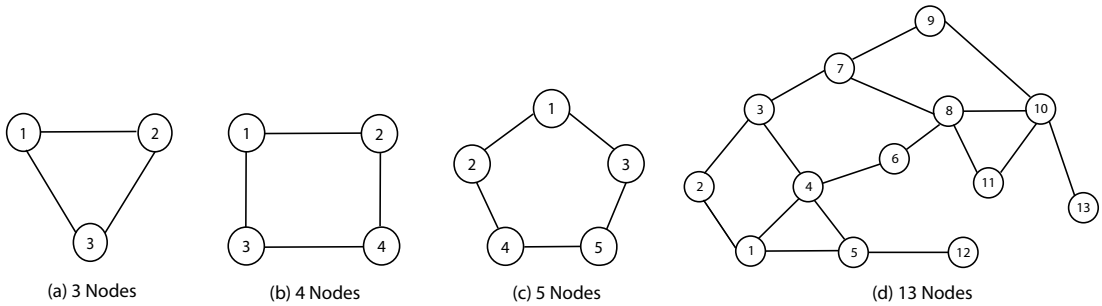

Figure 5.1: Samples of node connection

5.1.1 Three Node Network

For the three node network in Figure 5.1(a), it has a corresponding queueing model as in Figure 5.2. Since the state variable $x(t)$ is defined based on their destinations, such three node network will need a total of six differential equations; two equations representing each network node, namely: $x(t) = [x_1^2(t), x_1^3(t), ..., x_2^3(t)]$. To demonstrate the accuracy of the hybrid modeling approach, we have built a simulation of the time varying queueing behaviors of the network using standard simulators ns-2 and Qualnet. Throughout the simulation, the average payload size is fixed at 1250 bytes and 56 byte header being appended to each data packet as it goes through the protocol layers. All links have the same capacity (C_i) of 10^4 bps and the service rate at each node equals to $\mu = 1/1250$ bytes which is equivalent to a normalized server capacity of one packet per second. To demonstrate how traffic is rerouted through relay nodes when a link failure happens, we first show the simulation results on three node network. The rate of externally generated traffic for each node pair are set as $\gamma_1^2 = 0.18, \gamma_1^3 = 0.22, \gamma_2^1 = 0.18, \gamma_2^3 = 0.22, \gamma_3^1 = 0.18, \gamma_3^2 = 0.22$. By using the explicit Runge-Kutta (4, 5) formula, with variable small time step Δt, to solve our differential equations, we are able to calculate numerical solutions to the model in Equation (6), and compare our results to the ones determined from simulations.

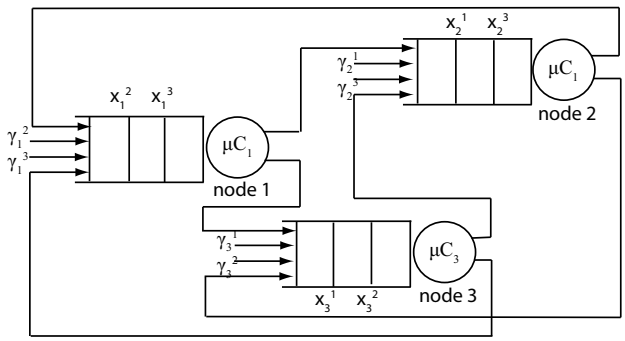

Figure 5.2: Queueing model for a three node network

5.2 HYBRID MODEL COMPARISON WITH SIMULATION

To account for time varying behavior, the nonstationarity component of the network comes from topology changes when link connectivity is being manipulated. A simple simulation scenario of three nodes with pre-determined connectivity changes between nodes at each time interval during the length of simulation time as illustrated in Figure 5.3(a) - (f) is used to evaluate the accuracy of our proposed model. In MANET, the link failure can be considered as node moving out of radio range and links are being disconnected. In this setup, we only allow one broken link at a time. When the direct link is no longer available, traffic must be rerouted through relay nodes and uses some available portion of the shared link capacity. Element $a_{ij}(t)$ in the adjacency matrix that represents node connectivity and routing control parameter $r_{ik}^{j}(t)$ must be calculated from routing algorithm at each corresponding time points. Here, we use minimum hop routing for the rest of the numerical experiments and simulations.

5.2.1 Numerical Results on Queueing Networks

Case I: Systems of Poisson Arrivals with Exponential Packet Size

Figures 5.4 - 5.9 show the effect of topology changes on the average number of packets of each traffic class at each node queue when Poisson input traffic has exponential packet length.

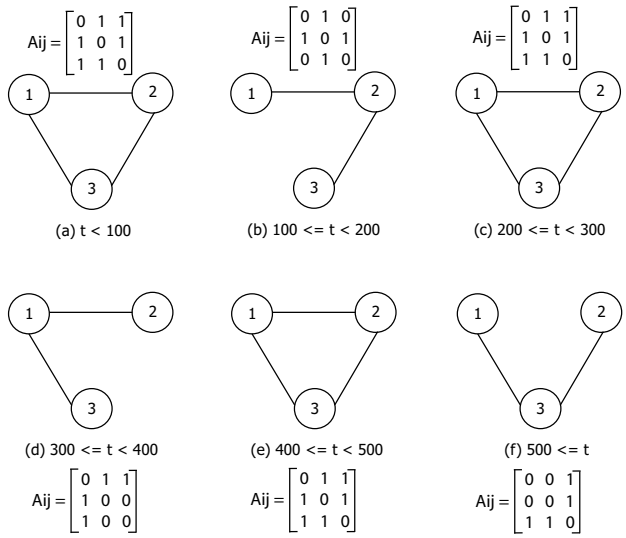

Figure 5.3: A connectivity scenario

For the time interval $t < 100$ sec, all nodes are connected and have direct links to send packets to the destination nodes. Nodes go through an initial transient period and finally reach the steady state. For time $100 \leq t < 200$ sec, the link between node 1 and 3 breaks and traffic going through this link has to go through the relay node. We can see the traffic being dropped from link 1-3 and being re-routed through link 1-2 and 2-3 instead, causing the increased number of packets in the buffer of $x_1^2, x_2^1, x_2^3, x_3^2$ during this time interval until the broken link is restored again at time $t = 200$ sec. For the time interval $300 \leq t < 400$ sec, this time the link between node 2 and node 3 breaks, leading to traffic re-routing and the increased number of packets in the buffers $x_1^2, x_2^1, x_1^3, x_3^1$. Similarly, a large increase in the number of packets in the system occurs at the relaying nodes when the link between node 1 and node 2 breaks during the time $500 \leq t < 1000$ sec. Figure 5.4 - 5.9 show that the results from the fluid flow model matches well with the simulation results, which means our proposed hybrid modeling approach is accurate.

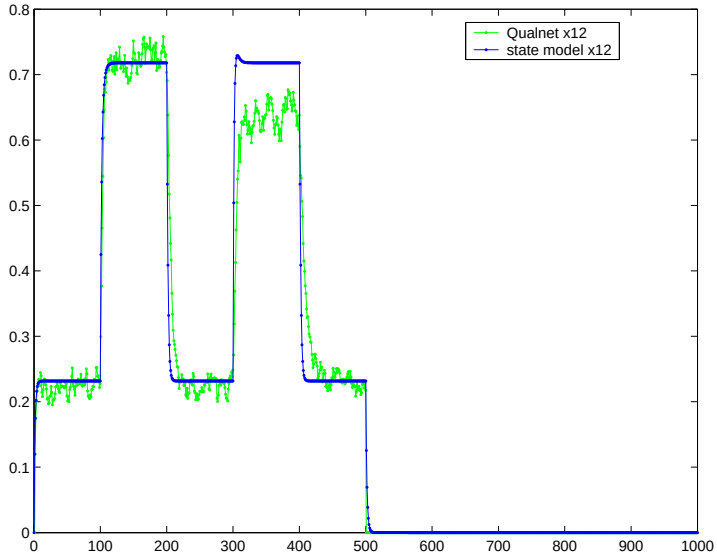

Figure 5.4: Number of packets x12 at node 1 buffer with exponential packet size

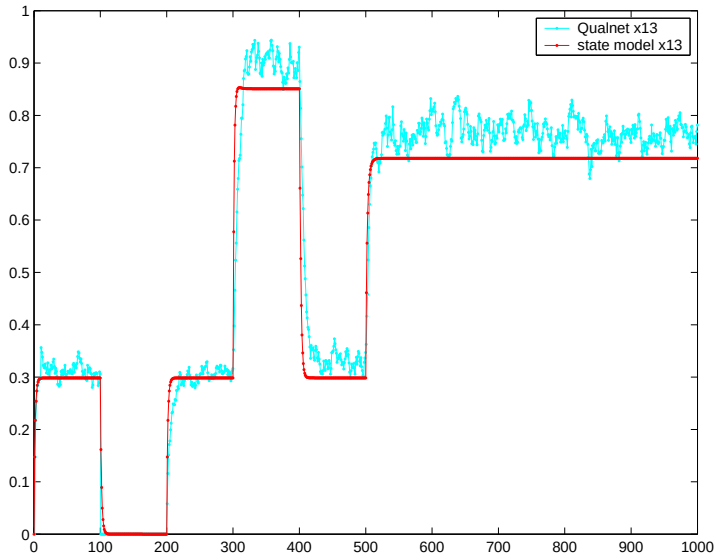

Figure 5.5: Number of packets x13 at node 1 buffer with exponential packet size

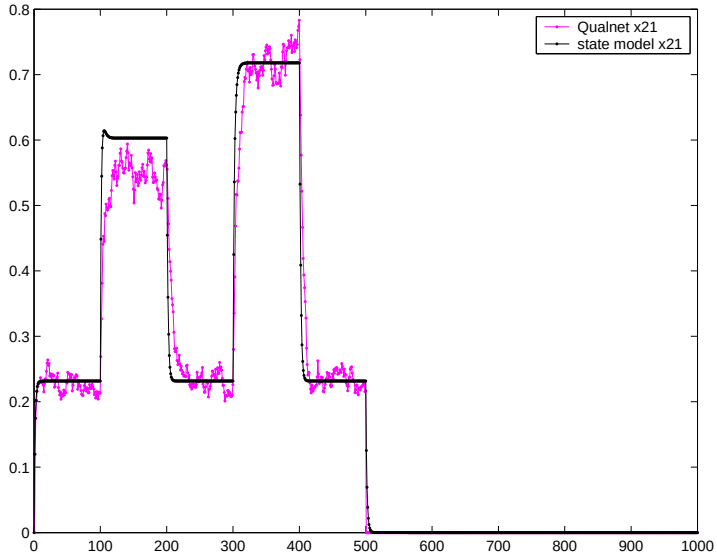

Figure 5.6: Number of packets x21 at node 2 buffer with exponential packet size

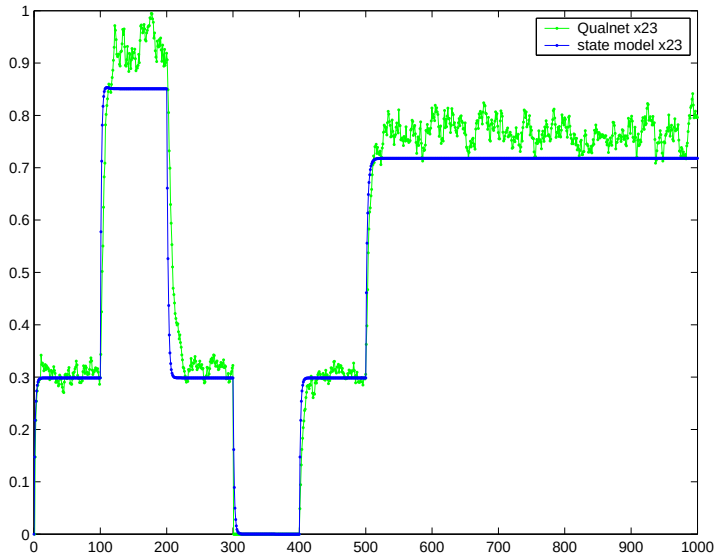

Figure 5.7: Number of packets x23 at node 2 buffer with exponential packet size

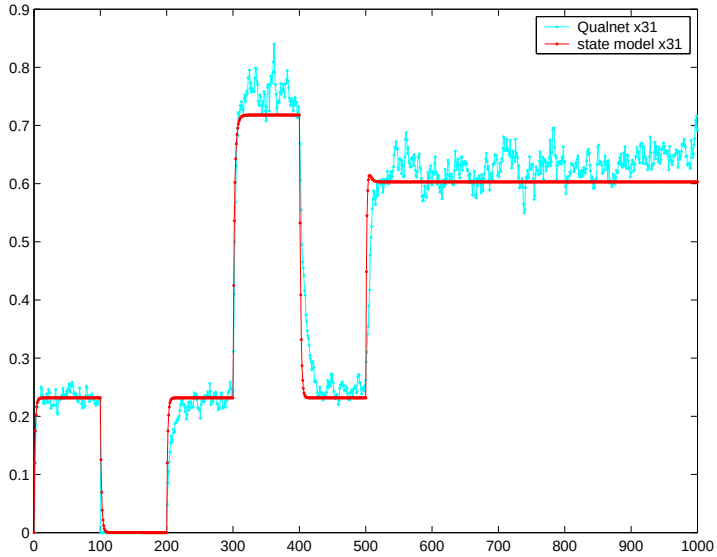

Figure 5.8: Number of packets x31 at node 3 buffer with exponential packet size

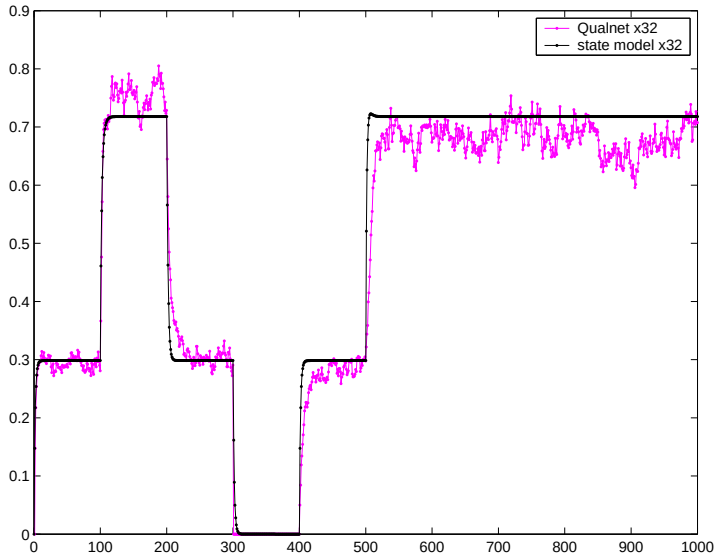

Figure 5.9: Number of packets x32 at node 3 buffer with exponential packet size

Case II: Systems of Poisson Arrivals with Constant Packet Size

Figures 5.10 - 5.15 are the results for the average number of packets of each traffic class at each node queue when Poisson input traffic has fixed packet length for the same three node simulation setup. The results show similar behavior in the sense that when direct link between two nodes is not available, traffic needs to be rerouted through relay nodes causing the large increase in number of packets at those node buffers during such time interval until the broken link is restored. From Figures 5.10 - 5.15 that the results from hybrid state model approach provides reasonably accurate outcomes which match well with the simulation results. From the results, the average number of packets at the node buffers in the case of constant packet size is generally less than the one with exponential packet size, which is consistent with known queueing theory results.

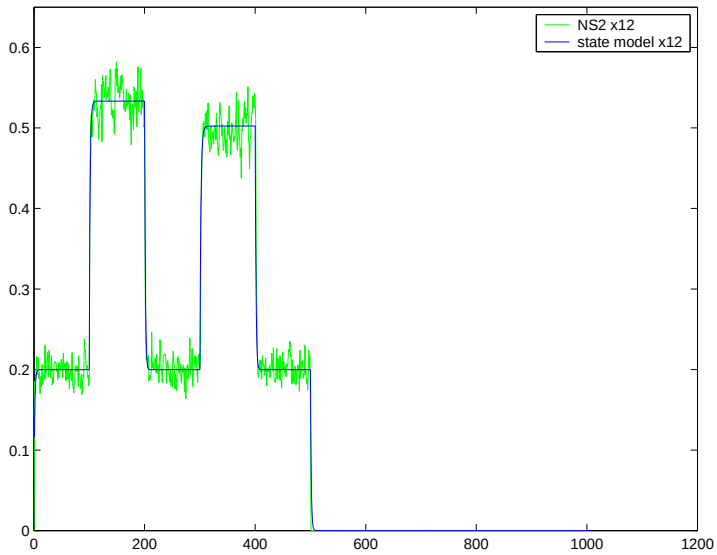

Figure 5.10: Number of packets x12 at node 1 buffer with constant packet size

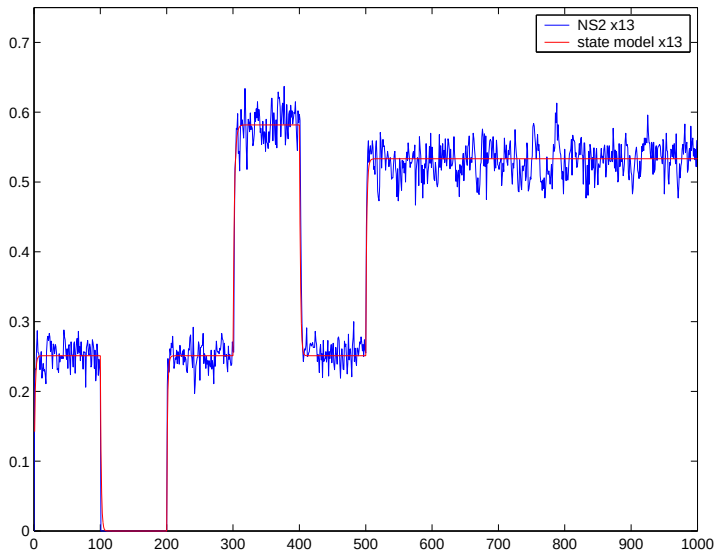

Figure 5.11: Number of packets x13 at node 1 buffer with constant packet size

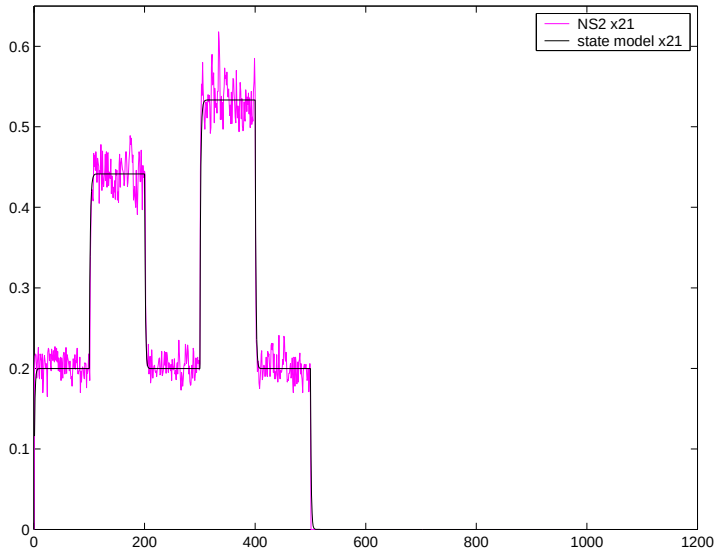

Figure 5.12: Number of packets x21 at node 2 buffer with constant packet size

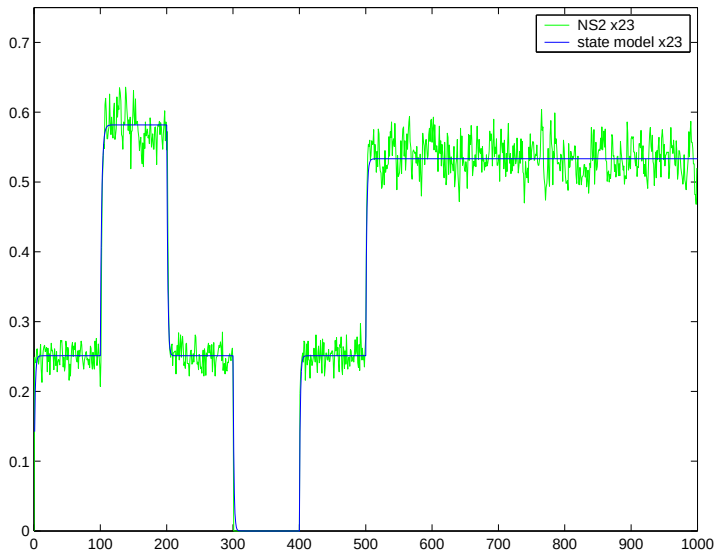

Figure 5.13: Number of packets x23 at node 2 buffer with constant packet size

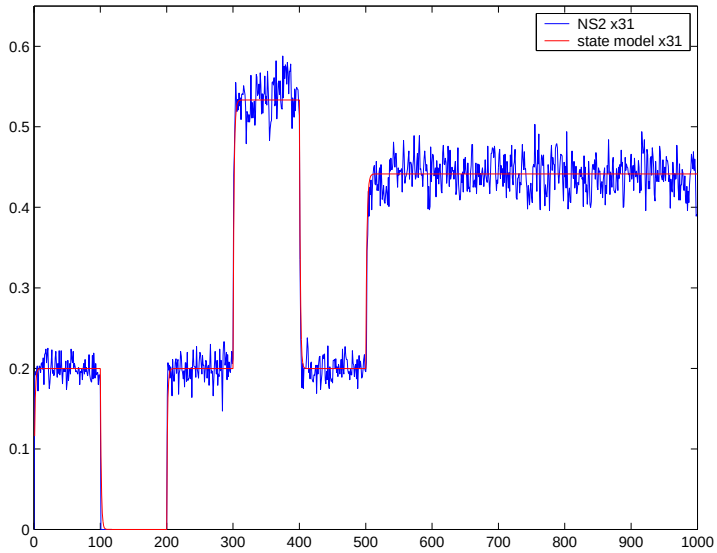

Figure 5.14: Number of packets x31 at node 3 buffer with constant packet size

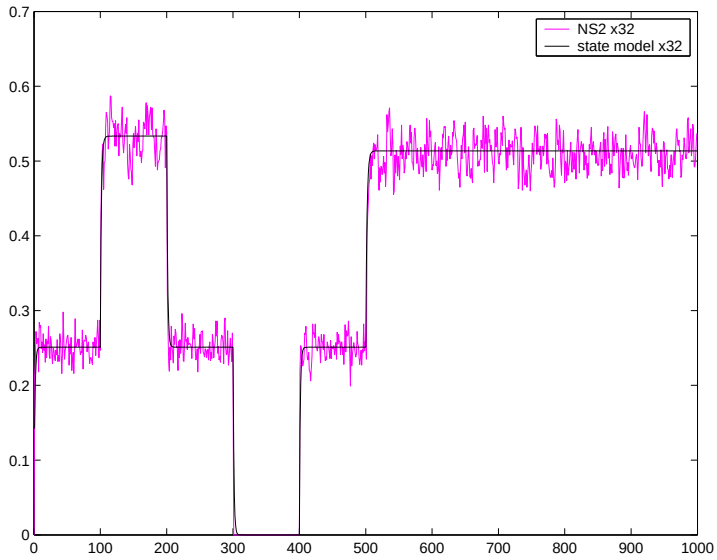

Figure 5.15: Number of packets x32 at node 3 buffer with constant packet size

Case III: Systems with the Superposition of Periodic Arrival Streams

Similar to the result of Poisson traffic, when the input traffic are periodic arrival streams, the results from fluid flow model also show the capability of being a good match with simulation results from the same scenario setup. For example, Figure 5.16 - 5.17 show the average number of packets at each node queue when periods of traffic from all source nodes are the same. Based on single link failure assumption, traffic aggregation is still less than service rate capability to ensure system stability. The rate of externally arrival packet for each node pair γ_i^j is set to be equal, and the server at each node processes data packets at specific deterministic rate. In this experimental setup, we test the accuracy of the proposed model with such arrival streams on the thirteen node network in Figure 5.1(d) by setting $\gamma_1^{10} = \gamma_2^{10} = \gamma_4^{10} = \gamma_5^{10} = 0.22$ and the service rate is being normalized to one packet per second. The nonlinear utilization function $G(x_l(t), x_T(t))$ in Equation (3.22) comes from data fitting approach.

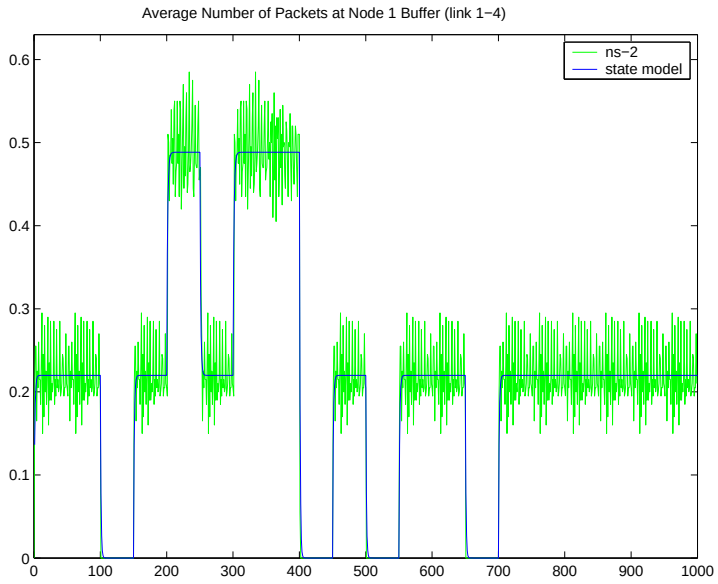

Figure 5.16: Number of packets x14 at node 1 buffer with periodic arrival streams

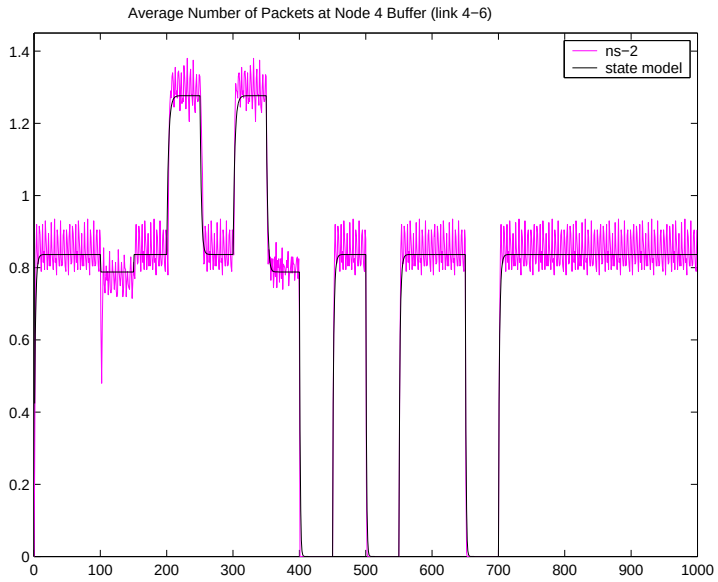

Figure 5.17: Number of packets x46 at node 4 buffer with periodic arrival streams

Figure 5.16 - 5.17 are only sample results, chosen as the busy link that traffic re-routing must go through when link failure occurs based on our simulation setup. Figure 5.16 shows the average number of packet x_1^4 at node 1 buffer as link failure is being introduced to the network topology. During the time $200 \leq t < 250$ and $300 \leq t < 400$, the resulting graph shows a large increase in the number of packets in the system occurs at the relaying nodes from traffic rerouting when any of the link 2-3, link 3-7 and link 4-5 breaks and some of the given input traffic needs to go through this alternate path during those specific intervals. Similar behavior applies to the average number of packet x_4^6 at node 4 buffer in Figure 5.17. We can see that in the case of $ND/D/1$, the proposed fluid flow model is still able to offer a reasonably accurate result even the utilization is not in a closed form as in a more commonly used $M/M/1$ or $M/D/1$ system and needs to be obtained case by case from data fitting approach.

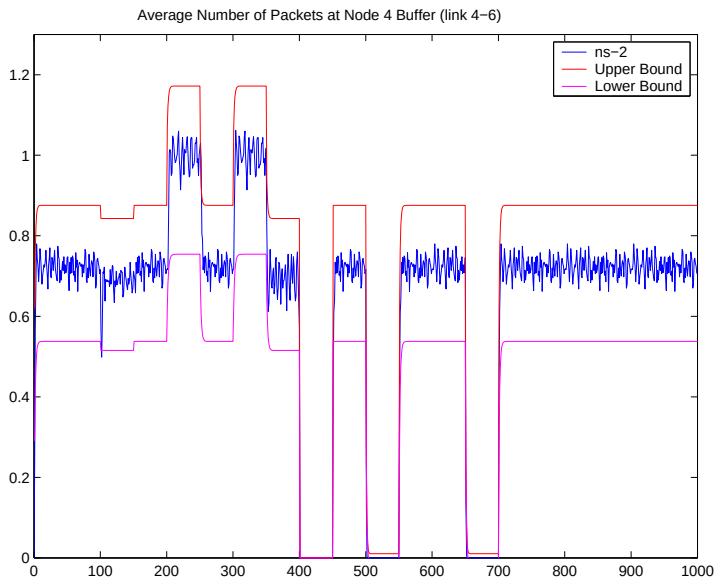

Figure 5.18: Average number of packets when periodic arrival streams have different periods

Finally, we tested the performance of the fluid flow model when input traffic are periodic arrival streams from non-identical sources, which is often referred to as a $\sum N_i D_i/D/1$ queueing system. The same assumption of single link failure still applies and total traffic aggregation is less than normalized service rate capability of one packet per second to ensure stable system. The rate of externally arrival packet is set as: $\gamma_1^{10} = 0.15, \gamma_2^{10} = 0.15, \gamma_4^{10} =$

$0.22, \gamma_5^{10} = 0.22$ and $\gamma_7^{10} = 0.25$ packets per second. Similar to the case of periodic input streams earlier, the general closed form of nonlinear utilization function $G(x_l(t), x_T(t))$ is impossible to derive; however it can be obtained in the form of nonlinear polynomial expression using data fitting approach. Figure 5.18 shows the resulting of average number of packet x_4^6 at node 4 buffer as topology changes. Again, during the time $200 \leq t < 250$ and $300 \leq t < 350$, we see an increase in the number of packets at the relaying nodes from traffic rerouting when any of the link 2-3 or link 3-7 breaks and some of the given input traffic needs to go through this alternate path during those specific intervals. Figure 5.18 shows the same network behavior as in Figure 5.18 with less total input traffic going through this particular network link. The difference is that the utilization function developed in this case will only provide the upper bound and lower bound of average packets in the system as shown in Figure 5.18.

5.2.2 More Experiments on Hybrid Models

a) End-to-end delay

Further experiments were conducted to show that the proposed hybrid model can be extended according to Equation (4.1) and Equation (4.2) to analyze the average time dependent queueing delay of the network. We tested the performance of the model on networks in Figure 5.1 with exponential packet size input and the same simulation setting as before while the propagation delay for each link (i, j) is assumed to be 0.01 seconds. In Figure 5.19, for time $0 \leq t < 100$ sec, the end-to-end delay of packet x_1^2 only comes from the propagation delay in link 1-2 and the queueing and processing delay at node 1. For time $100 \leq t < 200$ sec, link between node 1 and 3 breaks and traffic going through this link has to go through the relay node causing the increased number of packets at the node 1 buffer. As a result, the average queueing and processing time of each packet at node 1 will increase too and we can see the effect in higher end-to-end delay of packet x_1^2 during those specific time interval. On the other hand, link 1-2 breaks during the time $500 \leq t < 1000$ sec and packet x_1^2 has to go through other relay node in order to reach the destination. The communication path of packet x_1^2 contains 2 hops; therefore end-to-end delay will consist of propagation delay

in link 1-3 and link 3-2 as well as the queueing and processing delay at node 1 and node 3. Similarly, Figure 5.19 - 5.24 show the time dependent end-to-end delay of each type of traffic flow as the network topology changes. We also noticed the spike in simulation results from the hybrid model at the time instant where abrupt network topology change occurs and traffic is rerouted through that particular intermediate node. We assume that such incident is the consequence from the numerical integrator which should be solved with a proper choice of time step size.

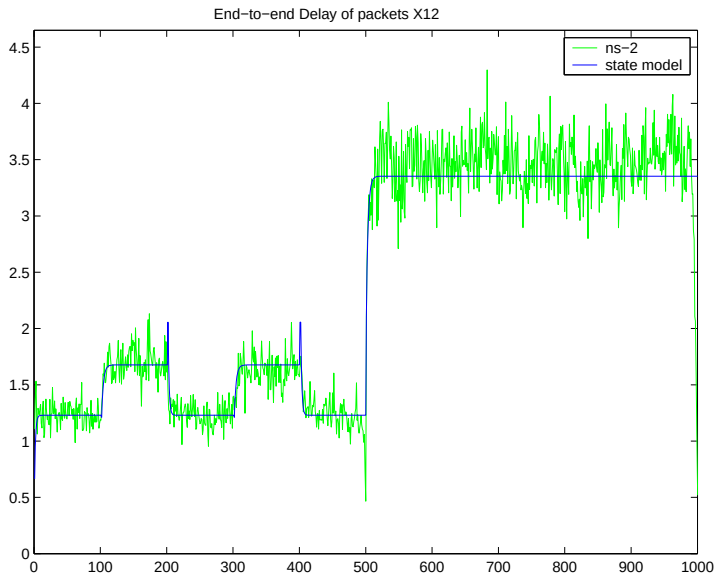

Figure 5.19: End-to-end delay for packet x12 with exponential packet size

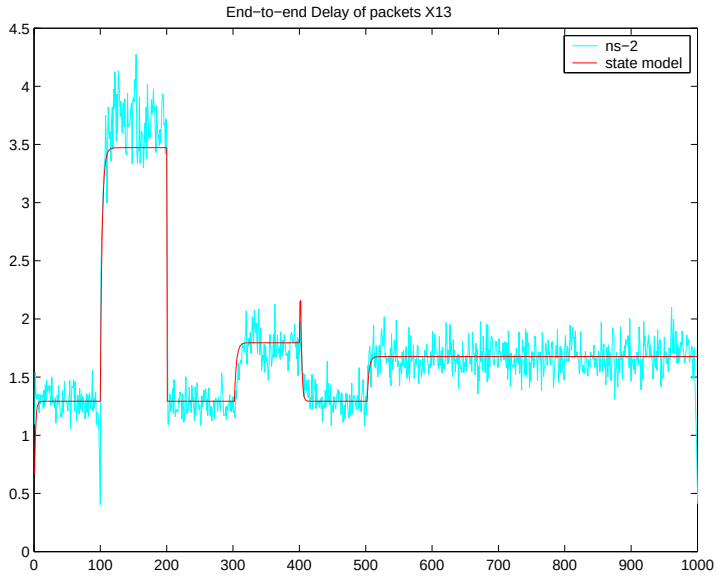

Figure 5.20: End-to-end delay for packet x13 with exponential packet size

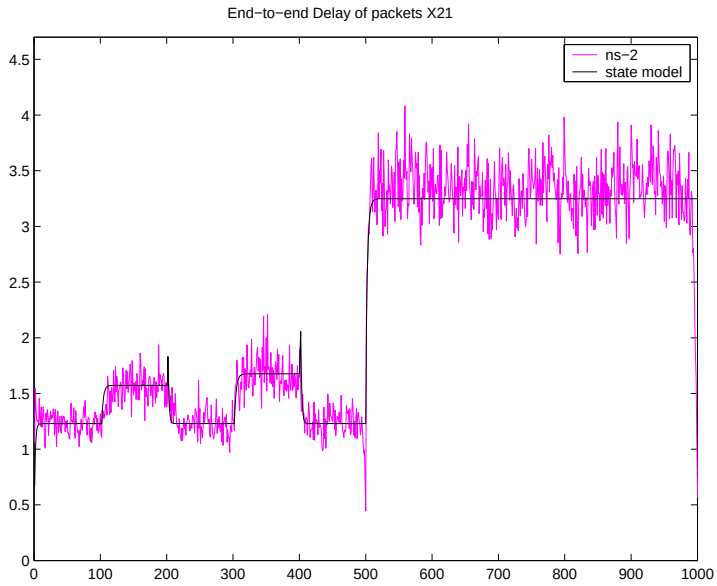

Figure 5.21: End-to-end delay for packet x21 with exponential packet size

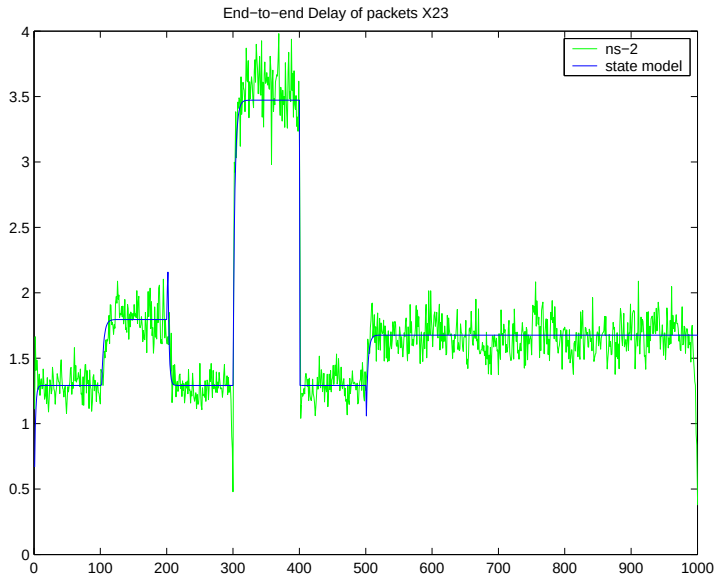

Figure 5.22: End-to-end delay for packet x23 with exponential packet size

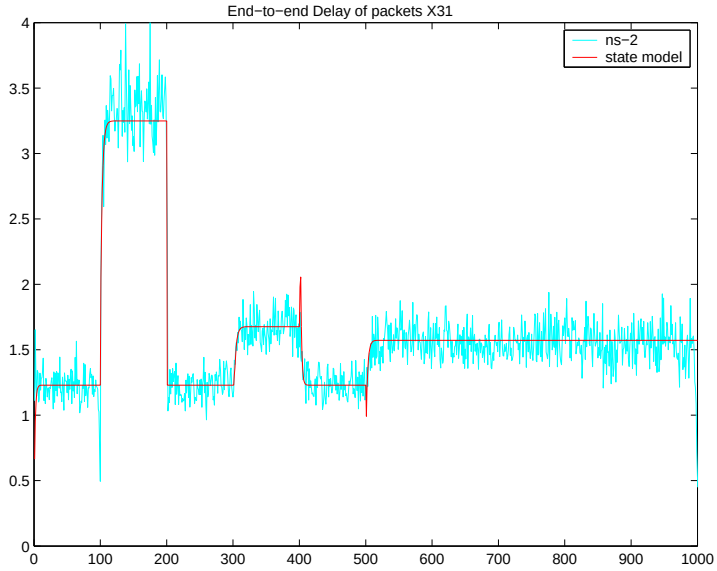

Figure 5.23: End-to-end delay for packet x31 with exponential packet size

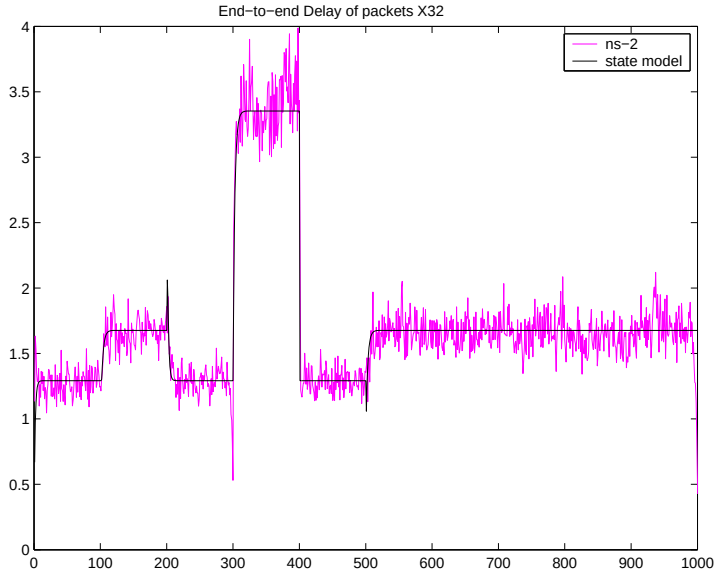

Figure 5.24: End-to-end delay for packet x32 with exponential packet size

To compare the results of average end-to-end delay when input traffic patterns are different, we set up the same simulation scenario as before - using three node network as in Figure 5.1 with link capacity of 10^4 bps to obtain the normalized server rate of one packet per section and arrival rates are set as $\gamma_1^2 = 0.18, \gamma_1^3 = 0.22, \gamma_2^1 = 0.18, \gamma_2^3 = 0.22, \gamma_3^1 = 0.18, \gamma_3^2 = 0.22$; however, each packet this time has the fixed size of 1250 bytes. We can see from the results shown in Figure 5.25 - 5.30 that each graph show less fluctuation, which is reasonable since packet length in this case is fixed while the variation in the previous results comes from the randomness of traffic with exponentially distributed packet size. However, the low fluctuation effect in the case of deterministic service queue was not included in the fluid flow model. Another finding from the simulation results is that with constant packet size, the service time required to process each packet will be deterministic, which leads to shorter queueing delay that the packets need to spend at each node buffer on average. With this reason, the graphs in Figure 5.25 - 5.30 shows shorter end-to-end delay than the results in exponential packet size case. Such results are consistent with what expected from the queueing theory.

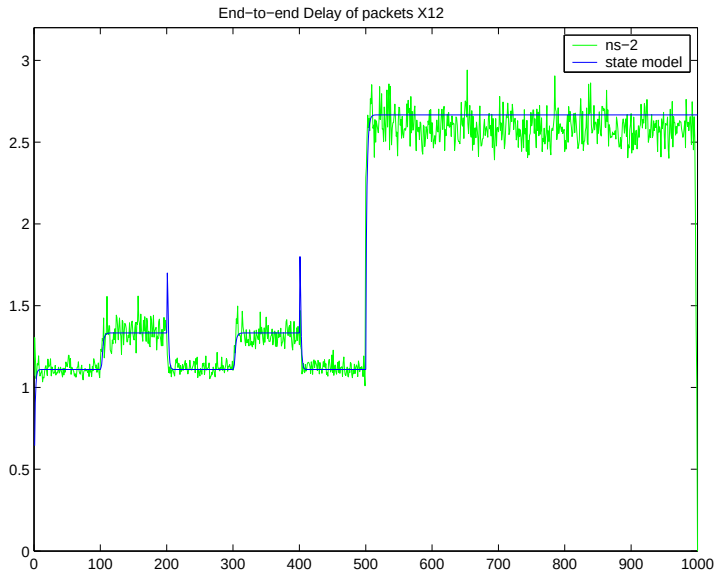

Figure 5.25: End-to-end delay for packet x12 with constant packet size

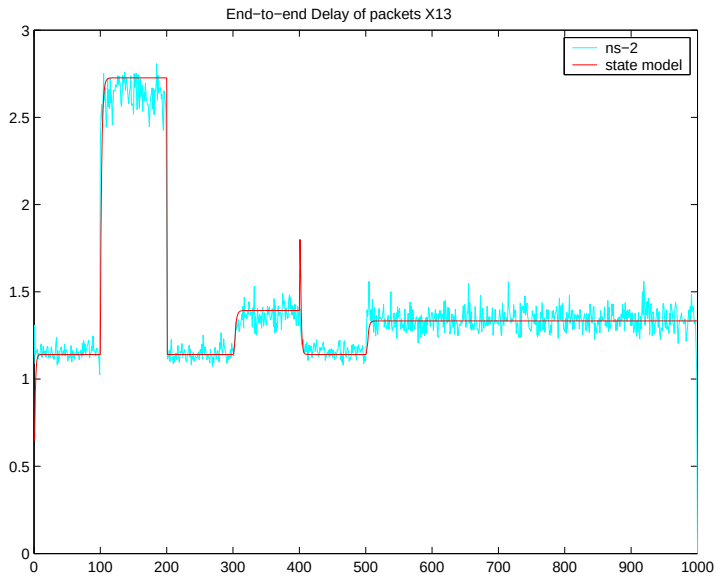

Figure 5.26: End-to-end delay for packet x13 with constant packet size

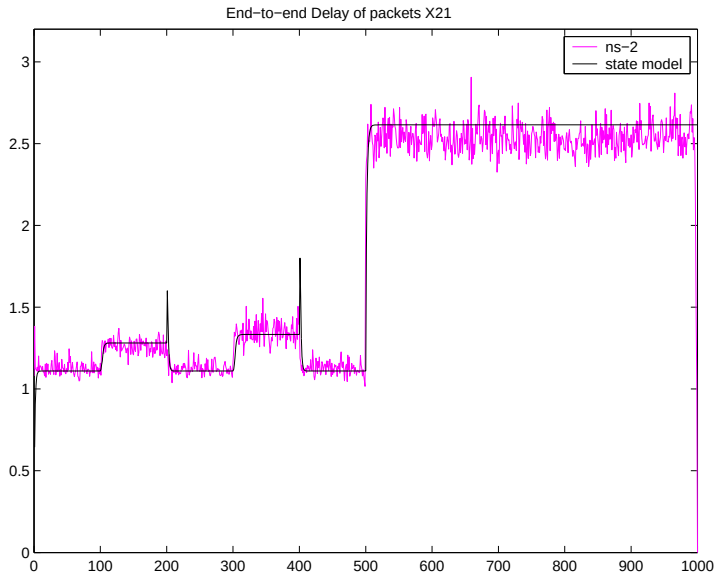

Figure 5.27: End-to-end delay for packet x21 with constant packet size

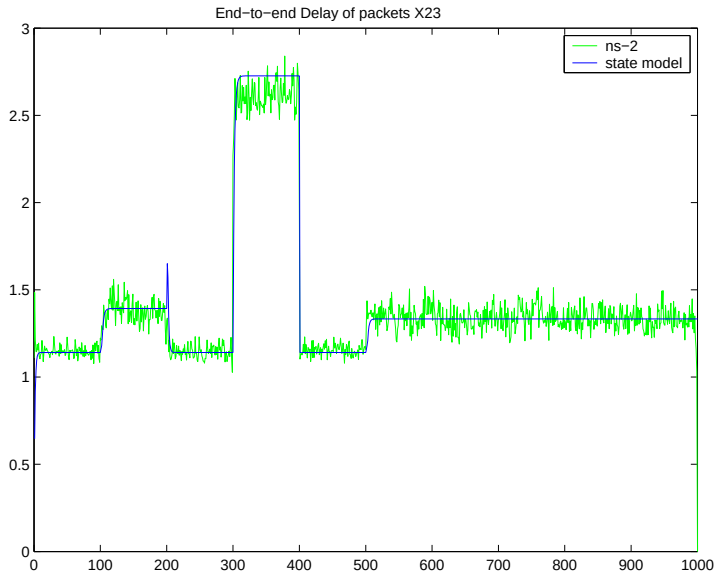

Figure 5.28: End-to-end delay for packet x23 with constant packet size

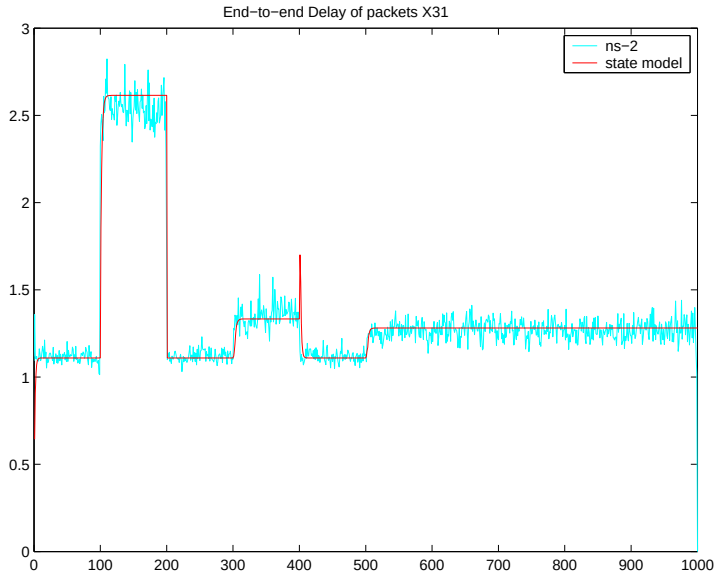

Figure 5.29: End-to-end delay for packet x31 with constant packet size

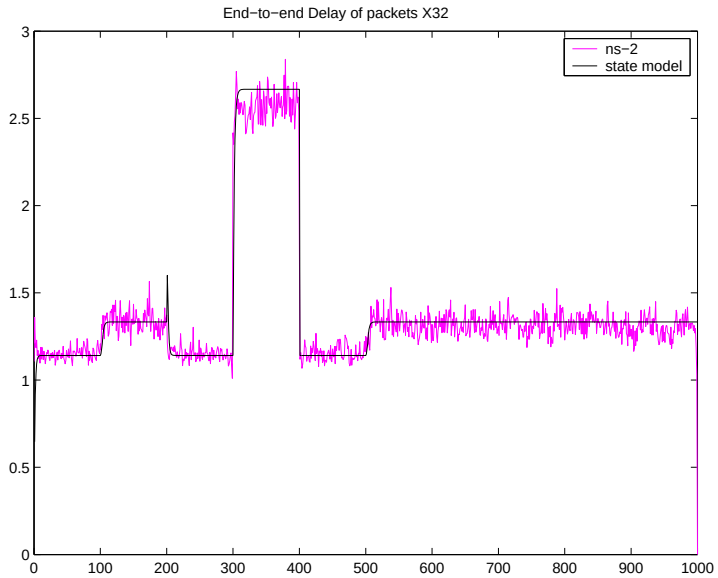

Figure 5.30: End-to-end delay for packet x32 with constant packet size

b) Packet delivery ratio

As discussed earlier in the previous chapter, we strongly believe that the way computer-based simulator simply returns one number as the packet delivery ratio to represent the overall performance metric of the whole system may not be appropriate especially for networks with nonstationary conditions. Further experiments were conducted to show that the concept of hybrid fluid-based performance model can be extended to evaluate the time dependent behavior of the packet delivery ratio by treating packet arrival rate and packet being delivered as data traffic flows. The link-level error is introduced to the the sample networks in Figure 5.1. Probability of link error is randomly assigned to each link in the network with error ranging from 1% to 10%. The performance in terms of number of packets successfully delivered at the destination compared to the total number of packets being sent out from the source node depends on factors such as the link quality and the policy on how to handle error packets; however, such performance measure does not depend on any stochastic modeling assumptions of the queue under study such as traffic arrival process or packet length distribution.

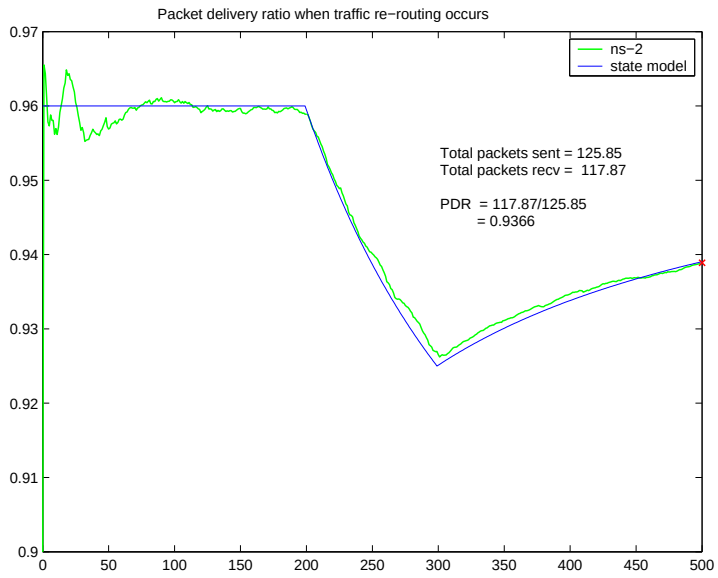

Figure 5.31: Time dependent behavior of packet delivery ratio (PDR)

Figure 5.31 shows the result in terms of time varying behavior of the packet delivery ratio on a simple three node network with probability of error assigned to each link. The link on the preferred path is broken during $200 \leq t < 300$ sec; therefore during that time period the traffic will be re-routed through links with higher probability of error until the broken link is being restored. Such changes will effect the number of packets received at the destination and lead to the lower packet delivery ratio. After the link is restored back to normal, traffic from source to destination node will use the path with lower probability of error to route the packets. Therefore, more packets will be successfully received at the destination, which implies the higher packet delivery ratio. Figure 5.31 shows that the result from the fluid model matches well with the simulation results and confirms that every incident happened in the network may effect the instant value of packet delivery ratio; therefore the packet delivery ratio should not be used as the overall performance metric.

5.3 COMPUTATIONAL SCALABILITY

In terms of computational time comparison, we tested the model performance on the three nodes, four nodes, five nodes, thirteen node shown in Figure 5.1 as well as the fifty nodes networks, with traffic generated from each node to all other nodes as a full mesh traffic to check on simulation time required for different network size. The simulations were run on a Sun Fire V240, 2x1.2 GHz UltraSPARC IIIi, 2GB machine and the numerical result is shown in Table 5.1. The result seems rational - the bigger the network, the longer the simulation time for both computer-based simulators and hybrid fluid-based analytical models. However, simulation time required by standard simulation tools seems to grow exponentially. The threshold for simulation time limit is set at 3 days and any simulation running longer than 3 days (approximately $260,000$ seconds) is considered indefinitely time-consuming and will be terminated prematurely, i.e., the result of the 50-node network as shown in the table. On the other hand, for the hybrid fluid-based model, as network size increases, the number of fluid flow differential equations to represent the network also increases, leading to longer computational time. However, when compared to the computational time required in standard simulators, the hybrid fluid-based model is considered a better scalable simulation tool.

Table 5.1: Computational time comparison in seconds

# Nodes	# Equations	Computer-based sim time	Hybrid model sim time
Three Nodes	6	5862.7	**0.20**
Four Nodes	12	6503.2	**0.46**
Five Nodes	20	11795.0	**0.81**
Thirteen Nodes	156	239,119.5	**4.40**
Fifty Nodes	2450	$>$ 3 days	**596.30**

Figure 5.32 is the result in terms of simulation time required to compute the metric of interest, namely the average number of packets in the system, as the network size increases. The result shows that when the network size is small, the computational time required is proportional to the number of differential equation with a certain rate. However, as the network size grows bigger, the computational time required to finish the simulation increases at a higher rate. We believe that with additional data points, it is possible to obtain a better curve fitting in polynomial expression, which is related to the computational complexity analysis, the big-O notation.

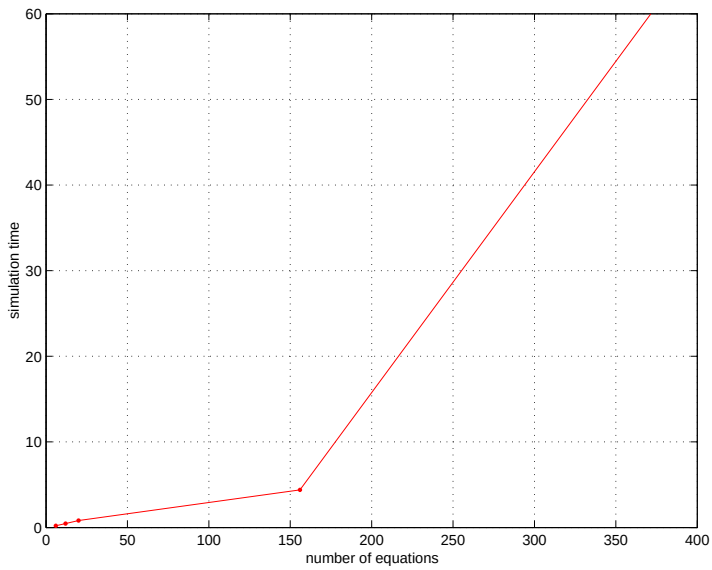

Figure 5.32: Computational Time vs. Number of Equations

6.0 CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

Mobile ad-hoc networks are getting more and more attention as many applications depend on its ability of nodes to coordinate and dynamically establish routes without any prior knowledge of existing infrastructure. Each node in the network acting like a router will figure a way to forward data packets to their destination by communicating with neighboring nodes within the radio range and providing a route, which may involve multiple hop communication through wireless links. The dynamic nature of the network topology comes from the arbitrary movement of each node, which when combined with all traditional problems of wireless communication make the design and development of mobile ad hoc networks more challenging.

Fundamental to the design of efficient mobile ad hoc networks is the ability to estimate and predict the network performance. However, mobile ad hoc network performance is normally studied via simulation over a fixed time horizon using a steady-state type of statistical analysis procedure. Simulation study of time varying behavior is possible, though computationally difficult even with the assistance from specialized simulation tools to reduce the complexity in the simulation modeling. As an alternative, network performance evaluation can also be done using analytical-based performance modeling technique. While the standard analytical network models mainly deals with steady state conditions, stochastic process can be used to describe dynamic network behavior under nonstationary conditions. The analysis will then involve solving a large set of differential equations and require a number of simplifying assumptions to make the model more manageable. Most of the previous works on the

performance of mobile ad hoc networks still use steady state analysis technique even with the presence of transient or nonstationary conditions in the networks. Therefore, the new performance modeling technique that is capable of analyzing the dynamic of the network, which includes both the time varying and steady state behavior is needed.

The background material related to network performance evaluation was presented in Chapter 2. Three basic commonly used techniques, namely the measurement-based, the simulation-based and the analytical based, with their drawbacks were briefly described. Data collected from the measurement-based or the simulation-based is random such that a large amount of runtime is required to assure independently generated input and to increase the credibility of statistical results. To study nonstationary networks, ensemble averages across multiple simulation runs are more appropriate than the time averages, which leads to problem in computational resources and scalability issue. On the other hand, if the analytical technique is chosen, it mainly deals with steady state conditions and the optimal steady state control parameters will be returns. To use analytical technique to model the time-varying behavior, the analysis now involves solving a set of differential equations and become difficult to obtain solution.

Approximation approaches can be used to capture the time varying behavior when dealing with the nonstationary conditions under the assumption that during the small segments of time interval, the steady state conditions are true. The definition, the advantages and the disadvantages of each particular approximation method were described. An early work using analytical based numerical scheme to study the nonstationary queueing model was presented as an example to show that solving a large set of differential equations for the state probability over each time interval is computationally complicated; thus settling with solution for transient behavior numerically is easier than deriving a closed form expression. Several numerical approaches including fluid flow model were proposed as a way to study nonstationary behavior of queueing system in terms of a more general performance measure, while providing more efficiency in computation and accurate results within reasonable bounds.

As the work in this dissertation focus on the mean nonstationary behavior of queuing system, a hybrid performance modeling technique was proposed. The model employs the concept of fluid model to describe the dynamics of the network, and the pointwise stationarity to numerically solve a set differential equations over each time interval. The general background on fluid flow model was provided in the last part of the literature review for both of the single class traffic and the multiclass traffic model.

In Chapter 3, we presented the detail of a proposed performance modeling technique for queueing analysis of networks with nonstationary conditions using a hybrid of discrete event simulation and numerical method techniques. The proposed model consists of two components: the network topology modeling part and the time dependent queueing behavior part. Node connectivity that represents topology changes is incorporated into the model using either discrete event simulation technique or stochastic modeling of adjacency matrix element. For the part of nonstationary queue, we further extend the basic fluid flow equations from Chapter 2 by deriving average server utilization function for each specific type of queue under study. The connectivity elements in the adjacency matrix depends on their radio range which is a function of many factors. The work in this dissertation started with the simplified problem that connectivity between two nodes is assumed to be symmetric and a function of distance only. However, the probability of link error was later introduced into the adjacency matrix elements to make the model more realistic. With the dynamic changes in network topology being reflected in adjacency matrix, the effect of those changes on network performance was studied through the time dependent queueing behavior at each node in terms of average number in the system using a fluid flow based set of differential equations, which can be solved with any standard numerical integration methods.

As mentioned earlier, the utilization function in fluid flow equations depend on stochastic modeling assumptions of the queue such as traffic arrival process and packet length distribution. We have derived the utilization functions for common cases in queueing theory: Poisson arrival with exponential packet size, Poisson arrival with constant packet size, and

the superposition of periodic arrival streams. The memoryless property and the traffic aggregation attribute of Poisson input process simplify the problem and make it possible to derive the close form expression of utilization function when the Poisson input traffic has either exponential or fixed length packet size. However, traffic with constant packet length will have deterministic service time and the output from such system should be treated as an delayed input to the next state. An appropriate delay also needs to be adjusted on the fluid flow differential equations to compensate for deterministic service time so that the fluid model can correctly estimate the average number at each node buffer.

Chapter 3 also provided the detail when the property of Poisson process no longer applies to the input traffic and the server utilization cannot be easily obtained in a close form expression. For example, the average queue length for periodic arrival stream arriving at a single server will only fluctuate between 0 and 1; therefore, the utilization function cannot be derived using the same approach as in Poisson process but rather through numerical analysis on queue length distribution. When input is the multiplex of constant bit rate streams from different sources, buffer requirement is usually chosen as the considered performance metric. The average number of packets in the system can alternatively be expressed using complimentary distribution function instead of the traditional probability density. With the information of traffic load and the total average number in the system at each steady state equilibrium, data fitting method can be used to obtain the utilization function in the form of a nonlinear polynomial expression. In addition, the function from data fitting process is not in the general close form that would be applicable to any periodic arrival streams but rather depends on case to case of arrival rate and service rate of the input traffic. The idea works well for periodic arrival input streams from identical sources; however, when arrival traffic comes from various sources with different periods, the exact formula for queue length distribution cannot be obtained and only the upper and the lower bounds on such distribution will be given instead.

In Chapter 4, two additional performance measures were introduced to show that the fluid flow based approximation method is a flexible analytical tool that is not limited to only

the average number in the system but can be extended to study other aspects of queueing networks. The end-to-end delay is chosen as a performance metric to analyze the latency in delivering the packet as it is being forwarded from the source via a path which may include multiple hops through several intermediate nodes until it reaches the destination. Compared to other types of delay, the queueing delay is considered the main factor in end-to-end latency of which its time dependent behavior can be studied using Little's theorem on fluid flow differential equations. In Chapter 4, the connectivity elements in adjacency matrix no longer represent perfect communication between two nodes, but rather were modified to be more realistic by including probability of link error. For system with loss, the packet delivery ratio is commonly used as an overall metric to analyze the network performance. The time dependent behavior is possible if treating traffic as data flows by applying the fluid flow model.

We have tested the performance of the proposed fluid model under different stochastic queueing assumptions on sample networks. The numerical results have been given in comparison and seemed to match well with the outcomes from standard computer-based simulators. The estimated performance metrics in terms of end-to-end delay and packet delivery ratio from the hybrid fluid-based model were provided and showed that it could be used as time varying performance measures to study the dynamic nature of nonstationary networks. Lastly, the computational time required by both approaches was shown side by side. We believe that this hybrid modeling approach is a proper tool for evaluating the time varying behavior of networks with nonstationary conditions. With the computational time saved from the fluid flow state model method, it is a tremendous gain in flexibility for modeling complex networks or developing other optional features of nonstationary effects to add higher level of fidelity into the proposed model.

As the work in this dissertation primarily focus on the queueing analysis of nonstationary networks, the results showed that the proposed hybrid fluid-based model is able to estimate and predict the network performance with reasonable level of accuracy. However, with the time limitation, there are still issues left for further elaboration in order to improve the

performance of the hybrid model. Figure 3.1 provides an idea of other key components that can be developed as the add-on features to make the hybrid model more complete. These issues are described in the next section as possible future work.

6.2 FUTURE WORK

6.2.1 Mobility Models

To this point, the hybrid fluid-based performance modeling technique developed in this work can only be used to evaluate the behavior of the network with stationary mobile nodes. This means that though the impact of link addition and link failure was included in the form of connectivity elements in adjacency matrix, the actual node mobility or comparable link connectivity model has not been implemented yet. As we are still in the early stage to develop and verify the accuracy of the hybrid modeling tool for dynamic behavior, modeling of mobility may add an additional source of error, which can be difficult to determine whether the discrepancies between the model and the real network come from the inaccuracies in the model or from errors in reproducing the mobility. However, we realize that to make our hybrid fluid-based model more realistic, this issue has to eventually be taken into account.

Node mobility can be implemented using random variables that describe speed and direction for various types of mobility characteristics [38]- [39]. Radio connectivity is calculated based on nodes' current speeds and directions, which can be computationally expensive. To have a better control of the relationship between the parameters of a mobility model and the changes in link connection especially for random waypoint, a probabilistic model using a two-state Markov chain to describe the duration of link being connected and the time when link being disconnected was introduced in [36], [37]. The probabilistic model offers computational efficiency even though the idea does not support high-fidelity mobility model or any physical layer characteristics. The study in [47] also confirms that the estimated

link duration is the best way to characterize the mobility; however, the simple exponential approximation of Markovian model will not be a good fit. In fact, link duration should not be independently determined but should be established along with the consideration of node velocity, pause time and total simulation time. A framework on the study of random way-point mobility was presented and can be used as a guideline to quantify the proper average link duration based on other mobility parameters to be used in the fluid-based model.

6.2.2 Hybrid Packet/Fluid Simulation

Since most of the standard simulators employ the concept of packet-by-packet model to represent any ongoing network activities through protocol stacks. This provides a better understanding of the overall network behavior by providing the detail of every packet in the network, but it typically comes at the cost of expensive and long execution time even for the reasonably sized networks. On the other hand, by treating data traffic as a flow and solving a set of differential equations to obtain statistical data, the fluid-based analytical model can be used to capture the dynamic behavior of the network as a whole, while providing the scalability as the network size increases. This section describes the possibility of future work to combine the standard simulators with fluid-based network performance modeling technique to take advantages of both methods.

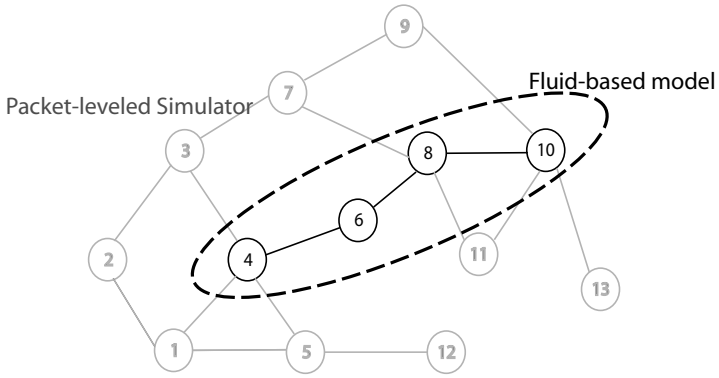

Figure 6.1: Hybrid packet/fluid simulator

As a minimum, the fluid-based model can be used together with the standard simulator

as a quick performance evaluation tool to see how the network performance may be effected by changes in the parameter setting. For example, let's assume that the path from node 4 to node 10 of the sample network in Figure 6.1 is the busy part of the network since most of the relay traffic need to go through node 4. If we would like to check how adding additional link can help improve the performance of the network in terms of average delay or number waiting in the queue, the fluid-based model is an appropriate performance tool for such scenario, rather than running the whole simulation with packet-level simulator again.

To take the advantage of being computational efficient and fully incorporate the fluid model into any standard simulators, an appropriate interface to handle data conversion between these two approaches has to be implemented, which may result in additional processing time. The input packets from data sources needs to be converted to traffic flows before entering the fluid flow section. Similarly, network statistics obtained from solving differential equations with the fluid flow model will be passed on to the standard simulators as normal incoming traffic. There are some research works regarding the integration of fluid and packet simulations; however the fluid flow modeling is only used as a tool to speed up simulations, not for the purpose of studying the dynamic behavior.

6.2.3 Models of Protocol Stacks

To send the data packet from source node to destination node, each packet has to go through protocol stacks, to which necessary overhead will be added. The effect of packet overhead can be added to the fluid model as the bigger average packet size, which then results in the lower effective service rate. However, in addition to the overhead, there are other elements regarding packets going through protocol layers that may have the consequence on the chosen performance metrics. For example, features in medium access control layer such as the inter-frame spacing structure, the random backoff time or the RTS/CTS mechanism can cause additional delay time since each packet has to wait at the node buffer for the control sequence, which will have effect on the delay related performance metrics including

end-to-end delay, time-varying average number in the system or throughput [48]. By taking these issues into account, the hybrid fluid-based model can be more complete and realistic.

6.2.4 Other Performance Metrics

In general, the performance measures can be considered either as (1) the computation time metric performance that is often used to assess speed efficiency or (2) the metrics that quantify simulation results and evaluate the result accuracy. We have shown that the hybrid fluid-based performance modeling technique, compared to the standard simulators, takes much less simulation time and can scale quite well as the network size increases. The speed up simulation implies less computational resource requirement or problem in scalability issue. Although the quantifying metrics that were used in this dissertation are sufficient to analyze the dynamic queueing behavior of nonstationary networks for our work, other possible performance metrics that may be more appropriate to investigate other aspects of network performance including throughput, the rate of packets being rejected by the network, the departure rate from the queue, or the total network link flow, are among possible candidates.

APPENDIX

SURVIVAL FUNCTION

A.1 THE MOMENT CALCULATION

Suppose X is a continuous random variable that takes on nonnegative values. Then we traditionally find the average value of such random variable from the expectation calculation using the following definition:

$$E[X] = \int_0^\infty x f_X(x) dx \tag{A-1}$$

where $f_X(x)$ is the probability density function of random variable X. This expression can be modified to represent a general case by defining $g(.)$ as a function of X. Then Equation (A-1) becomes:

$$E[g(x)] = \int_0^\infty g(x) f_X(x) dx \tag{A-2}$$

Instead of using the density of variable X to determine the average value of X as we normally do, it is possible to use the survival function of X as a replacement. Recall that:

$$S_X(x) = \Pr[X > x] = \int_x^\infty f_X(x)\, dx = 1 - F_X(x) \tag{A-3}$$

The survival function is the probability that the value of random variable X exceeds the value x, which can be expressed as the integral of the density on the interval $[x, \infty)$. The survival function is basically the complement of the cumulative distribution function $F(x)$.

Equation(A-2) can be further solved with by parts integration, with the choices of $u = g(x),\ du = g'(x)dx;\ dv = f_X(x)dx,\ v = \int f(x)dx\ = F_X(x)$. Therefore, using boundary conditions that $S(0) = 1$ and $S(\infty) = 0$, along with the assumption that $g(0)$ is finite or assumed to be zero as an initial condition, Equation (A-2) can be written as:

$$\begin{aligned} E[g(x)] &= \int_0^\infty g(x)f_X(x)dx \\ &= \big[g(x)F_X(x)\big]_{x=0}^\infty - \int_0^\infty g'(x)F_X(x)dx \\ &= \big[g(x)(1-S(x))\big]_{x=0}^\infty - \int_0^\infty g'(x)(1-S(x))dx \\ &= \int_0^\infty g'(x)S(x)dx \end{aligned} \tag{A-4}$$

As we focus on the average value, $g(x) = X$

$$E[g(x)] = E[X] = \int_0^\infty S(x)dx \tag{A-5}$$

In some cases, Equation (A-4) is more flexible and easier to work with compared to the original definition, especially when we need to obtain the k-th moment of variable X or when $g(x) = x^k$ for some positive integer k.

BIBLIOGRAPHY

[1] I. Chlamtac, M. Conti, and J. Liu, "Mobile Ad Hoc Networking: Imperatives and Challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, July 2003.

[2] D. Tipper, Y. Qian, and X. Hou, "Modeling the Time Varying Behavior of Mobile Ad-Hoc Networks," in *Proceedings of 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, October 2004.

[3] S. Sharma, "Approximate models of nonstationary queues and their application to communication networks," Master's thesis, Clemson University, 1992.

[4] *Glomosim Simulation Tool*, http://pcl.cs.ucla.edu/projects/glomosim/.

[5] *OPNET Simulation Tool*, http://www.opnet.com/.

[6] *NS-2 Simulation Tool*, http://www.isi.edu/nsnam/ns/.

[7] *QualNet Simulation Tool*, http://www.qualnet.com/.

[8] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET Simulation Studies: The Incredibles," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 50–61, October 2005.

[9] J. Heidemann, N. Bulusu, J. Elson, C. Intanagonwiwat, K. chan Lan, Y. Xu, W. Ye, D. Estrin, and R. Govindan, "Effects of Detail in Wireless Network Simulation," in *Proceedings of the SCS Multiconference on Distributed Simulation*, January 2001, pp. 3–11.

[10] D. Cavin, Y. Sasson, and A. Schiper, "On the Accuracy of MANET Simulators," in *ACM Workshop on Principle of Mobile Computing*, October 2002.

[11] W. Wang, D. Tipper, and S. Banerjee, "A Simple Approximation for Modeling Nonstationary Queues," in *Proceedings of IEEE INFOCOM 96*, 1996.

[12] D. Tipper and M. K. Sundareshan, "Numerical Methods for Modeling Computer Networks Under Nonstationary Conditions," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 9, pp. 1682–1695, December 1990.

[13] D. Logothetis and K. Trivedi, "The Effect of Detection and Restoration Times for error Recovery in Communications Networks," *Journal of Network and Systems Management*, vol. 5, pp. 173–196, 1997.

[14] D. Tipper, J. Hammond, S. Sharma, A. Khetan, K. Balakrishnan, and S. Menon, "An Analysis of the Congestion Effects of Link Failures in Wide Area Networks," *IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 179–192, 1994.

[15] M. Rumsewicz and D. Smith, "A Comparison of SS7 Congestion Control Options During Mass Call-in Situations," *IEEE/ACM Transactions on Networking*, February 1995.

[16] H. Shin, C. Charnsripinyo, D. Tipper, and T. Dahlberg, "The Effects of Failures in PCS Networks," in *Proceedings of Third International Workshop on the Design of Reliable Communication Networks (DRCN 2001)*, Budapest, Hungary, Oct., 7-10 2001.

[17] D. Tipper, C. Charnsripinyo, H. Shin, and T. Dahlberg, "Survivability Analysis for Mobile Cellular Networks," in *Proceedings Communication Networks and Distributed Systems Modeling and Simulation Conference 2002*, San Antonio, Texas, Jan., 27-31 2002.

[18] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks," in *Proceedings IEEE INFOCOM 2002, Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, April 2002.

[19] H. Lundgren, E. Nordstron, and C. Tschudin, "Coping with Communication Gray Zones in IEEE 802.11 based Ad hoc Networks," in *Proceeding of the ACM Workshop on Mobile Multimedia (WoWMoM 2002)*, Atlanta, GA, September 2002, pp. 49–55.

[20] D. Kotz, C. Newport, R. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental Evaluation of Wireless Simulation Assumptions," *Int'l Workshop Analysis and Simulation of Wireless and Moblie Systems (MSwiM 04)*, October 2004, ACM Press, New York.

[21] C. Newport, D. Kotz, R. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental Evaluation of Wireless Simulation Assumptions," *SIMULATION: Transactions of the Society for Modeling and Simulation International*, vol. 83, no. 9, pp. 643–661, September 2007.

[22] D. Wu, D. Gupta, S. Liese, and P. Mohapatra, "Quail Ridge Natural Reserve Wireless Mesh Network," in *ACM International Workshop on Wireless Testbeds, Experimental Evaluation and Characterization (WiNTECH), in conjunction with ACM MobiCom*, September 2006.

[23] D. Wu, D. Gupta, and P. Mohapatra, "Quail Ridge Wireless Mesh Network: Experience, Challenges and Findings," in *Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007 (TridentCom 2007)*, 2007, pp. 1–6.

[24] K. Pawlikowski, H. D. Jeong, and J. S. R. Lee, "On Credibility of Simulation Studies of Telecommunication Networks," *IEEE Communications Magazine*, pp. 132–139, January 2002.

[25] A. Boukerche and L. Bononi, *Simulation and Modeling of Wireless Mobile Ad Hoc Networks*. New York: IEEE Press and John Wiley and Sons, Inc., 2003.

[26] Z. Ji, J. Zhou, M. Takai, and R. Bagrodia, "Simulation and experiments: Scalable simulation of large-scale wireless networks with bounded inaccuracies," in *Proceedings of 7th ACM International WOrkshop on Modeling Analysis and Simuation of Wireless and Mobile Systems (SWiM 2004)*, Venice, Italy, 2004.

[27] W. Lovegrove, J. Hammond, and D. Tipper, "Simulation Methods for Studying Nonstationary Behavior of Computer Networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 9, pp. 1698–1708, December 1990.

[28] K. Alnowibet and H. Perros, *Communications and Computer Systems: A Tribute to Professor Erol Gelenbe*, J. Barria, Ed. Imperial College Press, 2006.

[29] K. A. Alnowibet and H. Perros, "Nonstationary analysis of the loss queue and of queueing networks of loss queues," submitted.

[30] L. V. Green, P. J. Kolesar, and A. Svoronos, "Some effects of nonstationarity on multiserver markovian queues systems," *Operations Research*, vol. 39, no. 3, pp. 502–511, May-June 1991.

[31] L. Green and P. Kolesar, "On the Accuracy of the Simple Peak Hour Approximation for Markovian Queues," *Management Sciences*, vol. 43, pp. 80–87, 1997.

[32] W. Whitt, "The Pointwise Stationary Approximation for $M_t/M_t/s$ Queues is Asymptotically Correct as the Rates increase."

[33] K. Alnowibet, "Nonstationary Erlang Loss Queues and Networks," Ph.D. dissertation, North Carolina State University, 2004.

[34] A. Odoni and E. Roth, "An empirical investigaton of the transient behavior of stationary queueing systems," *Operations Research*, vol. 31, no. 3, May-June 1983.

[35] W. Grassmann, "Transient solutions in markovian queueing systems," *Computer and Operations Research*, 1997.

[36] T. Lin and S. Midkiff, "Mobility versus Link Stability in the Simulation of Mobile Ad Hoc Networks," in *Proceedings of Communication Networks and Distributed Systems Modeling and Simuation Conference (CNDS)*, January 2003.

[37] S. K. Hwang and D. S. Kim, "Markov Model of Link Connectivity in Mobile Ad Hoc Networks," *Telecommunication System*, vol. 34, no. 1-2, pp. 51–58, February 2007.

[38] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, pp. 483–502, 2002.

[39] J. Yoon, M. Liu, and B. Noble, "Random Waypoint Considered Harmful," in *Proceedings of INFOCOM 2003*, San Francisco, April 2003.

[40] E. Modiano, J. E. Wieselthier, and A. Ephremides, "A Simple Analysis of Average Queueing Delay in Tree Networks," in *IEEE Transaction on Information Theory*, vol. 42, March 1996, pp. 660–664.

[41] M. J. Neely and C. E. Rohrs, "Equivalent Models and Analysis for Multi-stage Tree Networks of Deterministic Service Time Queues," in *Proceeding of 38th IEEE Annual Allerton Conference on Communication, Control, and Computing*, October 2001.

[42] M. J. Neely, "Exact Queueing Analysis of Discrete Time Tandems with Arbitrary Arrival Processes," *IEEE International Conference on Communication (ICC'03)*, vol. 4, pp. 2212–2225, 2003.

[43] M. J. Neely, C. E. Rohrs, and E. Modiano, "Equivalent Models for Qeueing Analysis of Deterministic Service Time Tree Networks," *IEEE Transaction on Information Theory*, vol. 51, pp. 3576–3584, 2005.

[44] J. W. Roberts and J. T. Virtamo, "The Superposition of Periodic Cell Arrival Streams in an ATM Multiplexer," *IEEE Transaction on Communications*, vol. 39, pp. 298–303, 1991.

[45] T. J. Hrabik, "Queue Length Distributions for Two-Stage CBR Multiplexing," in *IEEE ATM Workshop Proceedings*, May 1998, pp. 378–384.

[46] J. T. Virtamo and J. W. Roberts, "Evaluating Buffer Requirements in an ATM Multiplexer," in *GLOBECOM'89*, Dallas, Texas, November 1989.

[47] E. Casilari and A. Trivino-Cabrera, "A Practical Study of the Random Waypoint Mobility Model in Simulations of Ad-hoc Networks," in *Proc. of the 19th International Teletraffic Congress (ITC'19)*, Beijing, China, Aug. 29 - Sep 2 2005.

[48] H. Kim and J. C. Hou, "A Fast Simulation Framework for IEEE 802.11-Operated Wireless LANs," in *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, New York, NY, USA, June 10 - 14 2004.