Performance Modeling of the Wireless Internet

Pieter Kritzinger and Paolo Pileggi
Data Network Architectures Group
Computer Science Department, University of Cape Town
Private Bag, Rondebosch, 7700, South Africa
Email: {psk,ppileggi@cs.uct.ac.za}

Abstract: *IEEE 802.11 (or WiFi) networks are now well-established as the primary solution for delivering broadband services to metropolitan areas and rural communities. Furthermore, such networks are both easy to implement and efficient at providing communications in support of rural fire-fighting and similar emergency services. Moreover, in the developing world wireless networks can be rapidly deployed in rural areas, providing access to the Internet from public kiosks for educational and entertainment purposes. Part of the growing solution are Wireless Mesh networks, where peers communicate with each another and connect through a back haul network to the Internet. The back haul network, which connects to the Internet, can be one of a number of competing technologies, such as the increasingly popular 802.16 standard. Such a wireless network architecture is also referred to as the Wireless Internet. As these networks become increasingly more complex, modeling to evaluate the expected QoS plays a crucial role in the design process. In this paper we advocate a hierarchy of models which build upon an analytic multi-class queueing network model. Furthermore, we show the results of comparing an analytic model with simulations of the associated network, using inter-arrival time and packet distributions of measured Internet traffic.*

Keywords: Mesh networks, wireless Internet, Weibull distribution, log-normal distribution, performance modeling, multi-class queueing networks, simulation, rural communication.

I. Introduction

IEEE 802.11 (or WiFi) networks are considered the *de facto* standard for accessing the Internet from a variety of locations such as hotels, airports, conference centers, etc. However, the limited transmission range of 802.11 may be inadequate to provide coverage in a rural area and it seems likely that a combination of 802.11 and the 802.16 series of standards will be used for this purpose. In addition wireless mesh networks where peers communicate with one another and with a gateway on a back haul network may well be part of the future infrastructure for rural or urban communication. Vertically integrated 802.11 and 802.16 networks are commonly referred to as the Wireless Internet (WIN).

Given the plethora of wireless networks and their interconnection there are a number of questions regarding call admission control and the efficiency of routing protocols. All such questions depend on the QoS, or performance, of the network. In order to predict the performance of these networks one requires an appropriate prototype or model. Simulation models are an obvious option, however they become complex, difficult to validate and require substantial processor time as the network increases in size. Simulation models therefore do not scale well.

Clearly, the ideal model of a network with guaranteed QoS will have to take into account the full network, including such detail as routing, scheduling and different Media Access Control (MAC) schemes. The authors were unable to find any such model in the literature and decided to explore Multi-class Queueing Network (MQN) models as a foundation for the analysis of the WIN. We also compare these analytic results with the results from a simulation of the WIN.

In general, service disciplines in MQN models are not particularly sophisticated and do not appropriately model the detail of, for instance, the 802.11 DCF or 802.16 Connection Admission Control (CAC) mechanisms. Distributions are represented by their mean values only, as opposed to the full distribution used in a simulation. Stochastic models such as the Markov model by Bianchi [1] for the 802.11 DCF or by Niyato and Hossain [2] for 802.16 CAC exist in the literature for the behavior of individual nodes. Our objective is to make use of such models to compute the class dependent mean service time for the particular DCF or CAC algorithm.

Moreover, the authors wanted to explore the robustness of MQN analytical models in the case where the workloads are not exponentially distributed as all Markov [1], [2], or MQN model arrival times assume. It has been shown [3], [4] that the exponential distribution does not adequately fit the profile of Internet traffic.

We organized the paper as follows: In Section II we describe our concept of the Wireless Internet. In the same section we describe the analytic model we propose for modelling the entire network while Section II-C discusses the simulation model we built of the same sample wireless Internet. Apart from the abstraction any model is only as good as the accuracy of the workload model. We describe the part of Internet workloads necessary for our study very briefly in Section III and the parameters drawn from that are mentioned in Section IV. The experimental results are given in Section V.

II. Models of the Wireless Internet: Servers, Routers and Links

Throughout this study we make use of the example WIN illustrated in Figure 1. The number of nodes were deliberately kept small with 6 nodes only, including the Internet. It is believed that any larger topology will be an extrapolation of this model and will only serve to complicate the discussion.

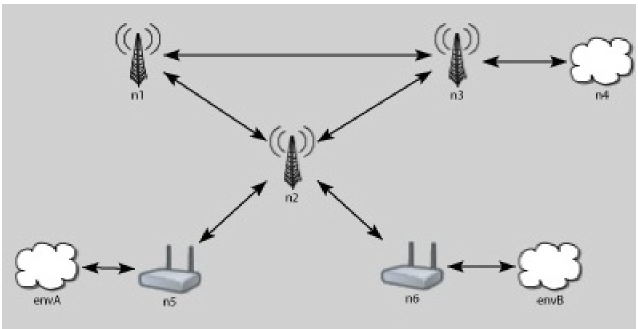

Fig. 1. Example Wireless Internet Network

The sample network consists of the following:

– 802.11 standard access points, called *Subscriber Stations* (SSs) (nodes N5 and N6 in the figure).
– 802.16 standard stations which are called *Base Stations* (BSs) (nodes N1, N2, N3 in the figure).
– A node, N4, modeling the delay in the Internet.
– Environments from which traffic arrive to the BSs, where traffic consists of IP packets carrying only web browsing traffic.

Subscriber stations communicate with the Base Stations using a version of the 802.11 standard and BSs communicate using a version of the 802.16 standard. In this model Subscriber Stations do not communicate directly, as they would in a mesh network. Such an extension to the model would be very trivial to incorporate. The specific version of the standard will be reflected in the sub-models and are not important for the base model.

A. Multiclass Queueing Networks (MQN)

Appropriate models for the network in Figure 1 are open multi-class MQN models. Multi-class MQNs have their origin in the classical work of Baskett *et al* (called BCMP networks) [5]. Closed or mixed (open and closed) queueing networks are also allowed if the modeling situation demands it.

In an MQN model provision is made for a number of different customer classes $r = 1, \ldots, R$, thus allowing each customer class to have a class-dependent service requirement from a server $i = 1 \ldots, N$. Each network class can typically represent a different 802.16 (or 802.11g) service class.

External class r customers arrive at node j according to a Poisson process with mean arrival rate $\lambda_{jr},\ j = 1, \ldots, N$ and $r = 1, \ldots, R$. Customers may change their class in going from one node to the next. That is, a customer of class r completing its service at node j may go to node i and change to class s with probability $p_{jr,is}$. Clearly, if $p_{jr,0}$ is the probability that a customer of class r will leave the network at node j, the routing probabilities satisfy:

$$\sum_{i=1}^{N}\sum_{s=1}^{R} p_{jr,is} + p_{jr,0} = 1 \quad j = 1, \ldots, N;\ r = 1, \ldots, R.$$

BCMP queueing networks allow for the following 4 different types of servers:

1) *First Come First Served (FCFS)* servers which must have exponentially distributed service times. In this case the mean service time may not be class dependent.
2) Servers with a *Last-Come First-Served Preemptive Resume (LCFS-PR)* rule. The service requirement of each class of customer may be arbitrarily distributed (see Chao *et al* [6]) and may depend on the class of the customer.
3) The third type of server is of the *Processor-Sharing (PS)* type and the service requirement of each class of customer may be arbitrarily distributed and may depend on the class of the customer. This basically means that, if there are n_{ij} class $i, \quad i = 1 \ldots, R$ customers present at server $j, \quad j = 1, \ldots, N$, and the service requirement of a class i customer is exponential with mean rate μ_{ij}, then the mean service completion rate at server j is
$$\frac{n_{ij}}{n_j}\mu_{ji}$$
where $n_j = \sum_{i=1}^{R} n_{ij}$.
4) A server with an *Infinite Service (IS)* capacity. Each customer, upon arrival, starts its service immediately and is delayed at the server for an arbitrary distributed service time with mean equal to that specified for the particular class to which the customer belongs.

Later work by Gelenbe [7] and others, described in Chao [6], adds *signals* to the network over and above regular customers. A signal may have a "customer" associated with it. When a signal arrives for service, it causes an event to occur. This event may be the addition (positive signal) or the deletion (negative signal) of one or more customers. The authors could not for the moment, see the application of signals in the models we developed. Internet traffic consists of down-link and up-link flows as discussed in detail in Section III. Up-link traffic are messages normally much shorter than down-link traffic. MQN models, because of their class dependent service time behavior (excluding FCFS servers) allow us to take this difference in service requirements into account.

The MQN used in this paper is illustrated in Figure 2. Customers of two down-link classes, shown as solid lines and each with its own unique service requirement, arrive at SS servers N5 and N6, respectively. We assume these servers to be of type PS since the assumption is that they use 802.11 (CSMA/CA), and typically the higher the traffic density generated by users in the immediate environment of these SS servers, the slower the throughput will be because of the contention resolution.

We assume therefore that all customers in the up-link are transmitting messages, since there are more than one channel available. Leaving servers N5 and N6, customers retain their class and enter the back haul network at BS server N2. Leaving N2 they are routed with probability $p_{2r,js}$ where $j = 1, 3$ and s = up-link $\vee$ down-link to either of servers N1 and N3. All three these servers are assumed to be generalized processor sharing servers as described by Parekh [8] and therefore represented as type PS in the MQN models. Note that in our model N1 has no external traffic arriving at it and serves merely as a relay station.

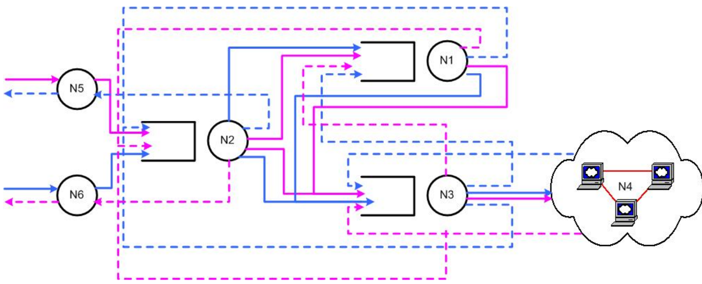

Fig. 2. MQN model of an example mesh network

All customers of the up-link classes which complete service at server N1 go to server N3 and all traffic go from N3 to the Internet, modeled by an IS server N4 which is assumed to impose a class dependent random delay on the traffic.

Having received an arbitrary distributed service time from the IS server, each up-link class changes to its respective down-link class (shown as broken lines) and returns in the same way as explained above, via the various servers, to their respective originating SS servers (as illustrated in Figure 2).

B. Solving the MQN models

Although the theory of MQNs was originally well-understood, it was not initially clear how to solve such models in finite time. Originally such models were solved using the so-called *Convolution Algorithm* invented by Jeff Buzen [9], but it was soon clear that problems with numerical stability arose in the solution of BCMP networks with implications for the accuracy of the answers.

In 1980 Martin Reiser and Steve Lavenberg [10] published the well-known Mean Value Analysis (MVA) solution technique. MVA was subsequently adopted by one of the authors [11] and his colleagues at the time, and turned into a software tool, called MicroSNAP [12], for solving open and closed MQNs. The interface to the tool is currently via a command language. The tool has been in use for many years and has withstood every stress test to date. MicroSNAP was used to solve the analytical models described in this report.

C. OMNET++ and Simulation

MQN assumes a Poisson arrival process. Whereas this may be an acceptable assumption, depending on the purpose of the model, the random variables mentioned seldom exhibit this ideal distribution. Nevertheless, most analyses, for example that by Niyato and Hossain [2], implicitly assume that this is the case. It was one of the purposes of our study to determine the error such an approximation would introduce. The authors therefore developed a simulation model, in which the arrival process is no longer Poisson. Furthermore, the simulation reflects the network shown in Figure 1 and not the MQN model.

For the simulation we used the OMNET++ (Objective Modular Network Testbed in C++) simulation development environment. OMNET++ provides a component architecture for models. Components (modules) are programmed in C++ and then assembled into larger components and models using a high-level language (NED). The environment uses a message passing model where the content of a message is defined by the user.

Modules are connected via gates (or "ports"), and combined to form compound modules. Connections are created within a single level of module hierarchy; a submodule can be connected with another, or with the containing compound module. Every simulation model is an instance of a compound module type. The components and topology are defined in NED files.

The simulator writes *output vector* and *output scalar* files. The capability to record simulation results has to be explicitly programmed into the simple modules by the model builder.

An output vector file contains several output vectors, each being a named series of (time stamp, value) pairs. Output vectors are capable of storing metrics such as queue length, end-to-end packet delay, packet drops or channel throughput over time according to how the simulation was programmed. It is possible to configure output vectors to enable or disable recording individual output vectors and limit recording to a certain simulation time interval.

III. Models of Wireless Internet Traffic

The difficult part of developing an analytical or simulation model is finding realistic values for the parameters of both the workload and system abstraction. One method is to measure such parameters by in an operational system. This is somewhat counterintuitive to the purposes of modeling, as the system does not yet exist in reality and, thus, no workloads can be measured.

In our study we assume that the traffic flowing into a WIN is IP traffic only, the largest proportion of which would be web queries on the up-link and web server responses on the down-link. There are a number of different measurements and analyses of web-traffic. Examples of such are those by Crovella and Bestavros [3], and work done at the authors' own institution by Walters [4]. The variables of interest to this study are the Web clients'

1) request Inter-arrival Time (IAT),
2) request size or size of up-link requests, and the
3) response size or down-link requests size.

The random variable distribution function that best fitted the inter-arrival time data is the *Weibull distribution*, which is given by

$$F(x;k,\alpha) = 1 - e^{-\left(\frac{x-x_0}{\alpha}\right)^k} \quad (1)$$

for $x \geq x_0$ and $F(x;k,\alpha) = 0$ for $x < x_0$, where $k > 0$ is the *shape parameter* and $\alpha > 0$ is the *scale parameter* of the distribution. The mean value of the two-parameter Weibull distribution is given by

$$E[X] = x_0 + \alpha\Gamma\left(\frac{k+1}{k}\right) \quad (2)$$

For the Web client request and response distributions the *log-normal distribution* provided the best fit. This is the probability distribution of any random variable whose *logarithm* is normally distributed and has a probability *density* function $f(x;\bar{x}_l,\sigma_l)$ given by

$$f(x;\bar{x}_l,\sigma_l) = \frac{1}{x\sigma_l\sqrt{2\pi}} e^{-(\ln(x-x_0)-\bar{x}_l)^2/2\sigma_l^2} \quad (3)$$

for $x > x_0$, where $\bar{x}_l$ and σ_l are the mean and standard deviation of the variable's *natural logarithm*. The mean value is given by

$$E[X] = e^{\bar{x}_l+\sigma_l^2} \quad (4)$$

There is no closed form for the log-normal cumulative distribution function.

In estimating how well various distributions fit large data sets, the λ^2 discrepancy statistic as defined by Pederson and Johnson [13], is normally used. Both the Pearson's χ^2 and the λ^2 statistics are based on histogram binning techniques and measure the magnitude of departure of empirical data from a mathematical distribution function fitted to the model. It has been found that for smaller data sets the λ^2 statistic was less biased and has smaller variance than the χ^2 statistic [13].

Of even greater importance was the fact that, unlike the Anderson Darling statistic, the λ^2 statistic could be used on large data sets.

Another advantage of the λ^2 statistic was that it could be used to compare the goodness-of-fit of tests performed on data sets with different sample sizes. It is not possible to compare tests performed on data sets with different sizes when using χ^2 or the Anderson Darling statistic. The λ^2 statistic may be used to compare results from tests performed on data sets of different sizes, as the sample size and number of bins are taken into account in the calculation of the statistic. The smaller the value of λ^2, the better the fit of the data to the function it is tested against as mentioned by Walters [4] (page 74).

IV. Parameter Values

Since we had access to the detailed analyses, we used the parameter values for IP traffic measured by Walters [4] and confirmed by Choi [14] and Barford [15]. The values found by Walters were of the same magnitude as those of the other authors.

Table I shows the mean and standard deviations measured by Walters and Choi respectively, for the web client parameters used in our models.

Parameter	MEAN ($\bar{x}$)		STD (σ)	
	Choi	Walters	Choi	Walters
Request Size (bytes)	360	418	107	156
Response Size (bytes)	7 758	5 222	126 168	15 994
IAT (milliseconds)	900	1 500	2 200	5 700

TABLE I

TABLE OF MEASURED PARAMETER VALUES

Both Choi and Walters found that the web client response and request sizes, respectively, were best approximated by a log-normal distribution, Eq. 3. A visual representation of the curve fitting can be seen in Figures 3 and 4 (Figures 36 and 42, respectively in [4]). Quantitatively, the λ^2 values for the various distributions and parameters are given in Table II. We included the values for the exponential distribution for comparison purposes only.

In the multi-class MQN model, as in the simulation, we distinguish between the service times of up-link and down-link traffic. MQN models allow this since customers can change class.

We used the distributions with the smallest λ^2 used in our OMNET++ model experiments described below. The mean and standard deviation used to compute the parameters (cf Eq. 3) were taken from Table I. Choi reports that the Gamma

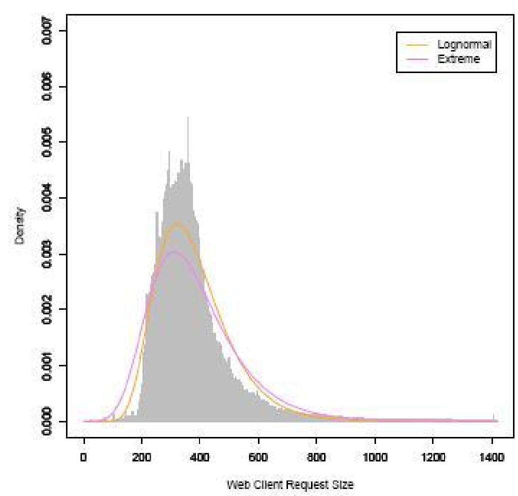

Fig. 3. Best fit comparisons for Web Client Request Size

distribution fits the web client request IAT best, while Walters determined the best (or better said, the best of the worst) to be the Weibull distribution. Figure 5 (from [4], Figure 32) gives a visual representation of the original data fitted to the different distributions. The quantitative difference (how well the two distributions fit the measured data as measured by the value of χ^2) is slight and we decided to use the Weibull distribution for the web client request IAT. Using the Gamma distribution would have been just as easy.

We also assumed an equal number of request arrivals at each of the two SSs in both the MQN and OMNET++ models. When varying the mean IAT in each simulation experiment,

	Exponential	**Weibull**	**Lognormal**	**Gamma**
Request Size (bytes)	1.722 (1:72; 1:725)	18.125 (17:894; 18:356)	0.766 (0:753; 0:778)	4.304 (4:251; 4:357)
Response Size (bytes)	38.723 (38:185; 39:26)	1.383 (1:348; 1:417)	0.104 (0:104; 0:105)	7.909 (7:744; 8:075)
IAT (milliseconds)	396.152 (393:44; 398:864)	0.046 (0:046; 0:046)	0.032 (0:032; 0:033)	0.97 (0:949; 0:991

TABLE II

TABLE OF λ^2 GOODNESS-OF-FIT VALUES

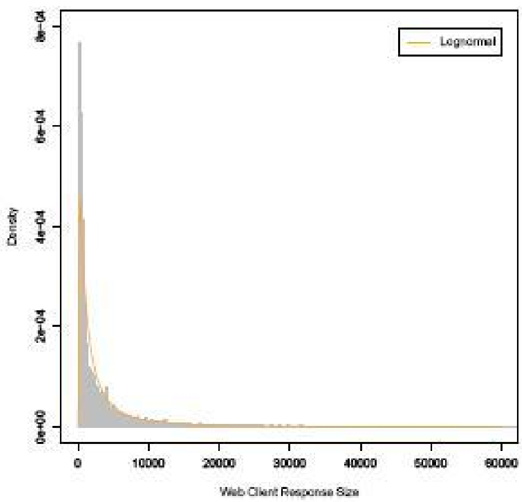

Fig. 4. Best fit comparisons for Web Client Response Size from [4], Figure 42

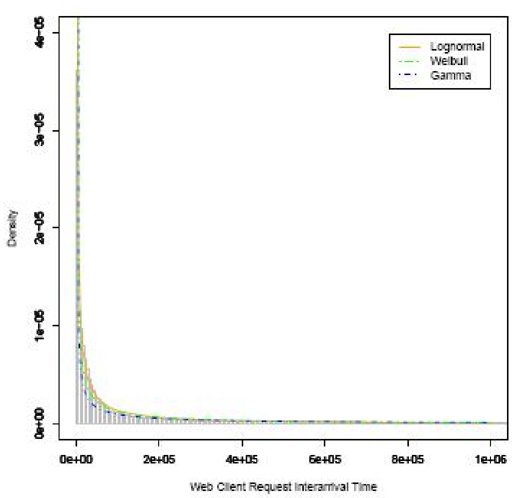

Fig. 5. Best fit comparisons for Web Client inter-arrival time distributions

we held the shape parameter i.e., $k = 0.371$ in Eq. 2) constant and computed the corresponding α value. In all cases we used mean IAT values which would stress the capacity of the network in order to amplify the difference between the analytic and simulation values.

An important parameter value that we did not know, and which Walters did not measure, is the Internet response time modeled by node N4 in either Figure 1 or 2. However, numerous measurements exist and one could choose any relevant value.

We selected arbitrary values of 1Mbps for the mean service times at a 802.11 node and 1.4Mbps at an 802.16 node. Mean request sizes on the up-link were chosen to be 488 bytes and 5222 bytes respectively (see Table I).

Whenever IP requests may be routed to more than one node, for instance at the AP node N1, they are assumed to go to each node with equal probability. This is an arbitrary assumption and is a parameter of either type of model and may easily be modified if measured values are available.

V. EXPERIMENTAL RESULTS

With the types of distributions and the values of the parameters chosen, we were able to experiment with both the MQN and the OMNET++ simulation models. From Figure 1 it should be clear that the most utilized node will be the AP node N2, which is therefore the node for which we recorded the queue length results reported below.

A. Validation

The first scenario assumes exponential distributions for all variables in the models. This is the simplest Coxian distribution (assumed to approximate the arbitrary distributions of MQN) and is done to validate the analytic versus the simulation models.

The results are illustrated as plots `OM: all exponential` in Figures 6 through 7 for the response time of a SS (either node N5 or N6) and Figures 8 and 9 for the queue length at the BS node N2. The analytic results are annotated with the letters `MS` in the figures and the simulated results with `OM`. 90% confidence levels are shown only for the plots with the Weibull IAT and log-normal distributions described later.

As can be seen from the figures, the error is relatively small — about 5% for the response time and almost 0% for the queue length measurements. The greater error in the response time arises from the difference between the way it is recorded in the simulation and the analytic tool which calculates the so-called *residence time* of a customer in the network.

B. Weibull Inter-arrival Times

In the second scenario we changed the IAT distribution in the simulation from exponential to Weibull using the parameter values suggested by Walters [4] from the analysis resulting in the fit illustrated in Figure 5. The results are illustrated as the plots `OM: Weibull IAT/log-normal message size` in the figures already mentioned.

The simulation model now reports a longer response time than in the first scenario. The results are fairly accurate

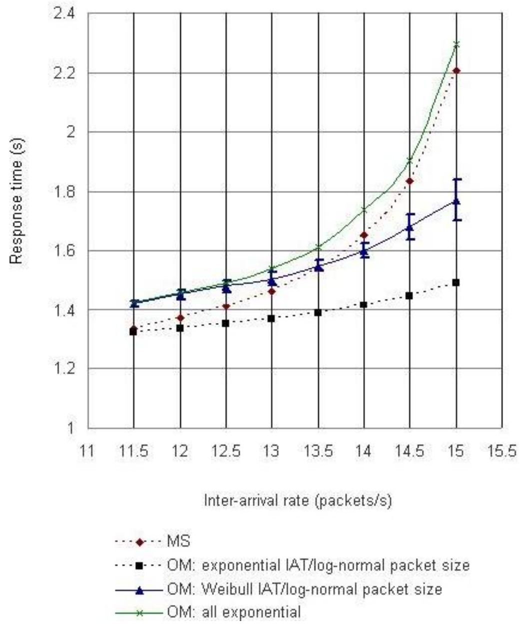

Fig. 6. Network response time

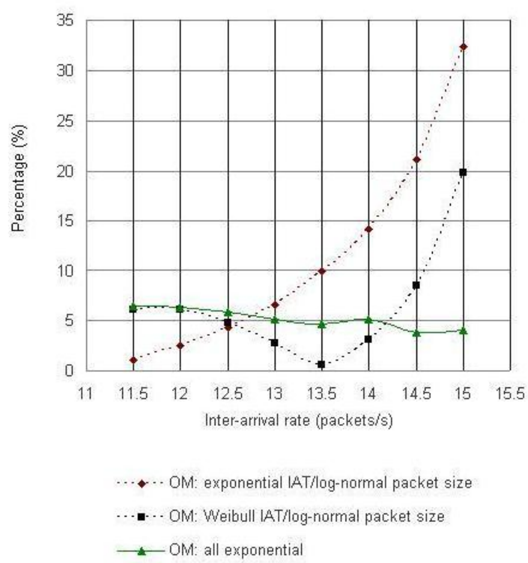

Fig. 7. Error between mean network response time simulated and analytic results

except at high arrival rates with an average error between the analytic and simulated response time values of 12% increasing from 0% percent to a high of 33% at saturation. The trend in the mean queue length results is the same except that the error is unacceptably high, with a mean of 48%. The Weibull distribution is a heavy tail distribution, with a higher probability than the exponential distribution of long IAT. It is to be expected that the queue length will be overstated in the case of the exponential distribution.

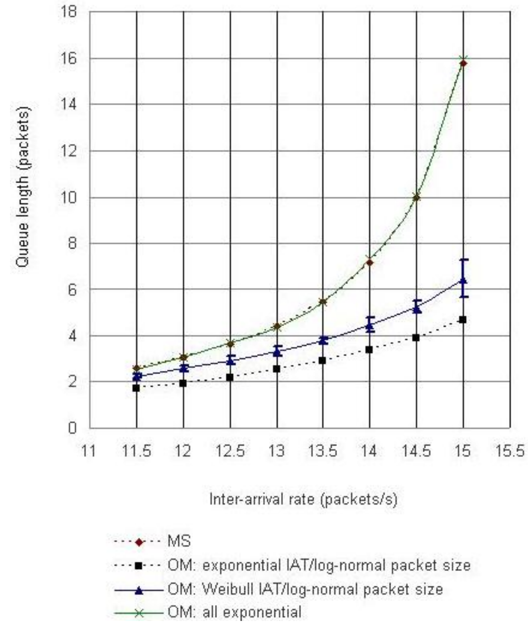

Fig. 8. Mean queue length at BS N2

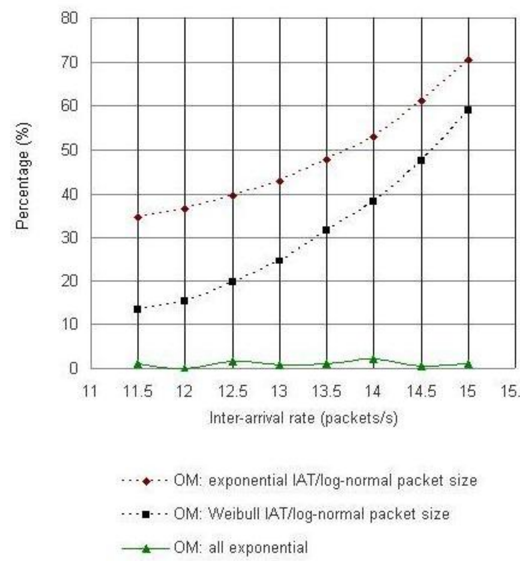

Fig. 9. Error between mean queue lengths at BS N2 for simulated and analytic results

C. Log-normal Service Times

In the last scenario we change the simulation model so that *both* the IAT and the message size distributions correspond to the measured behavior. The results are illustrated as the plots `OM: Weibull IAT/log-normal message size` in Figures 6 and 7 for the response time of a SS (either node N5 or N6) and Figures 8 and 9 for the queue length at the BS node N2.

The analytic values lie outside the 90% confidence interval except for very low arrival rates. At network saturation the difference between the mean response times reported by the analytic and the simulation models at node N2 is an acceptable 6% to 20%, as a percentage of the analytic value. For the mean queue length at node N2 the difference ranges from about 12% to a high 60% at saturation, which is unacceptable.

In concluding this section, the authors acknowledge that both the parameter values and the distributions used in their models may not accurately represent *every* typical web browsing scenario, since every session is likely to be different from all others. It is also likely that, due to the different behavior of wireless versus fixed line networks, the characteristics of the variables or their mean values and standard deviations will be different. As far as we know, no such measurements are available.

However, if anything, we wanted to base the parameter values on measured data and we do not believe that our assumptions invalidate the comparison of the results of MQN and simulation models.

VI. Conclusion

In this project we investigated whether an analytical MQN model with detailed sub-models would represent the WIN well enough for comparing the effect on QoS of various MAC scheduling or CAC schemes.

The motivation being that we do not believe it suffices to study the effect on QoS of a particular MAC, routing or CAC scheme at an isolated node. As the number of nodes increase, analytic models scale more easily and are less prone to errors introduced by a simulation.

The results show that the analytical results suffice to predict trends, which is probably all that is required if one wishes to compare the effect of different network configurations or MAC schemes.

Some of the differences between the analytic and simulation results reported can be attributed to different interpretations of the metric in either model. For instance, the MQN model solution uses Little's law to calculate the mean queue waiting time whereas in the simulation it is measured directly.

We have not yet included any sub-models, such as those presented by Niyato and Hossain [2] or Bianchi [1], in the results shown. We foresee this as the next step and expect to study the effect of changes at a detailed level on *overall* network performance.

Acknowledgement

The authors would like to thank Andrew Symington for his diligent reading of and the proposed editorial changes to the first draft of the manuscript and to a referee for suggesting a restructuring of some of the sections.

References

[1] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[2] D. Niyato and E. Hossain, "A queueing theoretic and optimization-based model for radio resource management in IEEE 802.16 broadband wireless networks," *IEEE Transactions on Computers*, vol. 55, no. 11, pp. 1473–1488, November 2006.

[3] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *Performance Evaluation Review*, vol. 24, no. 1, 1996.

[4] L. Walters, "A web browsing workload model for simulation," Masters, Department of Computer Science, University of Cape Town, Available at http://pubs.cs.uct.ac.za/archive/00000137/, May 2004.

[5] R. M. F. Baskett, K.M. Chandy and F. Palacios, "Open, closed and mixed networks of queues with different classes of customers," *J of the ACM*, vol. 22, pp. 248–260, 1975.

[6] M. M. Chao X. and M. Pinedo, *Queueing Networks: Customers, signals and product from solutions.* John Wiley and Son, 1999.

[7] E. Gelenbe, "Product-form queueing networks with negative and positive customers," *J of Applied Probability*, vol. 28, no. 565–663, 1991.

[8] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 334–357, June 1993.

[9] J. Buzen, "Computational algorithms for closed queueing networks with exponential servers," *Communications of the ACM*, vol. 16, no. 9, pp. 527–531, September 1973.

[10] M. Reiser and S. Lavenberg, "Mean Value Analysis of closed, multichain queueing networks," *J of the ACM*, vol. 27, no. 2, pp. 313–322, April 1980.

[11] P. Kritzinger and S. van Wyk, "Mean Value Analysis: a collection of the results," University of Stellenbosch, Tech. Rep. ITR82-14-00, April 1982.

[12] S. Crosby *et al.*, "Microsnap user manual," University of Cape Town, Department of Computer Science, Tech. Rep., August 1990, version 4.

[13] S. P. Pederson and M. E. Johnson, "Estimating model discrepancy," *TECHNOMETRICS*, vol. 32, no. 305, 1990.

[14] H. Choi and J. Limb, "A behavioural model of web traffic," in *7th International Conference on Network Protocols, Toronto, Canada*, October 1999.

[15] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," in *Performance 1998/ACM SIGMETRICS*, 1998, pp. 151–160.