Sizing of IEEE 802.11 Wireless LANs

Preetam Patil
Department of CSE
Indian Institute of Technology-Bombay
Mumbai- 400076, India
pvpatil@iitb.ac.in

Varsha Apte
Department of CSE
Indian Institute of Technology-Bombay
Mumbai- 400076, India
varsha@cse.iitb.ac.in

ABSTRACT

Sizing of IEEE 802.11 wireless LANs (WLANs), defined as the problem of finding the maximum number of users that can be supported, is essential for efficient application performance over WLANs. The usage of existing analytical models of 802.11 MAC in sizing requires a mapping from application load and performance to link-layer load and performance respectively, which we propose in this paper. We first evaluate analytical models of 802.11 MAC from the sizing perspective and then propose an approximate sizing method. We illustrate our method through an HTTP application and validate it through extensive *ns-2* simulations which show that the number of users suggested by our tool are within 13% of those derived from simulations for a majority of the test cases.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Performance

Keywords

Wireless LANs, MAC, Analytical Models, Simulations

1. INTRODUCTION

The effect of contention in wireless LANs (WLANs) based on IEEE 802.11 standard varies with operating parameters such as the number of users (nodes), the packet arrival rate at each node and average packet size. In this work, we concentrate on selection of one such parameter, i.e., *the number of users*. The problem, in the context of WLANs, can be defined as follows: given the usage profile of applications, find the maximum number of such users that can be supported within a basic service set (BSS).

WMASH'05, September 2, 2005, Cologne, Germany.

The use of analytical models can result in simple tools for answering sizing questions. Such a tool would take as input the application usage profile along-with WLAN parameters and produce sizing recommendations. Deriving an exact model that takes into account the behavior at the application, TCP, MAC and physical layer is a complex task; in the absence of such a model, we need to look at approximate approaches. The usage of existing analytical models for 802.11 medium access control (MAC) requires the specification of the traffic offered to the WLAN interface.

In this paper, we address the requirement of the translation of application usage profile to WLAN traffic profile, and a reverse mapping from the MAC performance (predicted by analytical models) to the application performance. We do this by first surveying existing analytical models, and comparing them with extensive simulations, to identify the model most suitable for sizing. We then derive a mapping between the application and the MAC by characterizing the application traffic. Our mapping makes simplifying assumptions and hence is approximate. We can see from the validations that despite these assumptions, the method works quite well. We illustrate our method with HTTP traffic, but it is general enough to be used with other applications (and their combinations), provided that the traffic offered by the application can be characterized.

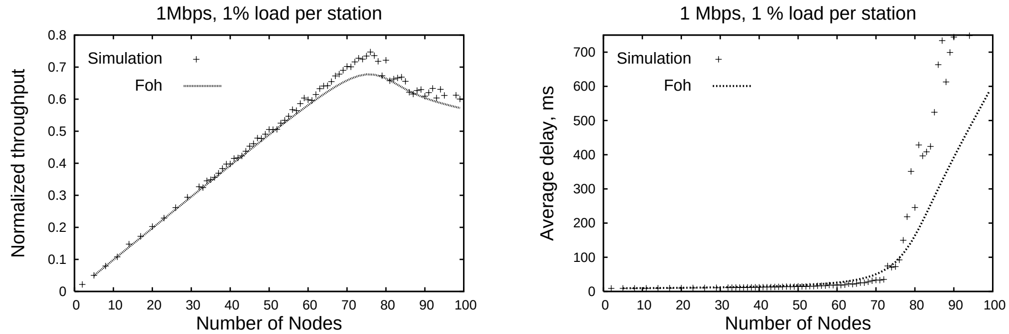
2. PERFORMANCE MODELS FOR 802.11 WLANS

Application performance over 802.11 WLANs has been studied mostly through simulations [2] and measurements [5]. File transfer times over WLANs have been studied analytically in [6] without modeling the TCP behavior. Analytical models for 802.11 MAC in saturated conditions have been proposed by Bianchi [3] and Tay-Chua [7]. Although saturation throughput is a fundamental measure of capacity, saturation models can not be directly used for sizing. An analytical model for obtaining non-saturation throughput and delay has been proposed by Foh-Zukerman in [4].

2.1 Comparison and validation of Analytical models for 802.11 MAC

We identified the models by Foh-Zukerman [4] and Tay-Chua [7][1] to evaluate their suitability for use in sizing. We validated these models through extensive simulations using

[1]Although the model in [7] is a saturation model, the authors also propose a closed form expression for the value of n^*, the saturation point.



(a) Throughput vs. n (b) Average delay vs. n

Figure 1: MAC performance from simulation and Foh-Zukerman's model

Pkt Size (bytes)	DataRate (Mbps)	%load /station	n^*_{Sim}	n^*_{Tay}	n^*_{Foh}	n^*_8
1000	1	1	80	64	75	84
1000	1	2	40	33	37	41
1000	11	1	46	43	47	46
1000	11	2	24	23	21	23
500	1	1	72	62	67	76
500	1	2	37	30	33	39
500	11	1	32	33	31	32
500	11	2	17	17	13	17
50	1	1	33	24	23	33
50	1	2	17	13	9	17

n^*_{sim}: simulations with single-packet Poisson arrivals
n^*_{Tay}: Tay-Chua's model
n^*_{Foh}: Foh-Zukerman's model
n^*_8: simulations with bursts of 8 packets

Table 1: MAC Congestion points n^*

the *ns-2* simulator [1]. We configured identical nodes to send traffic to each other in a logically circular topology, over an error-free channel, using the exponential traffic generator. Fig. 1(a) and Fig. 1(b) compare MAC throughput and average delay respectively, obtained from simulations and Foh-Zukerman's model for 1000B payload.

Table 1 lists values of n^*, the number of stations for which saturation sets in, obtained from simulations and analytical models. The per-station load is expressed as a percentage of the channel capacity. Based on the the numerical analysis, we conclude that Foh-Zukerman's model is able to predict throughput and delays accurately and is the most suitable for sizing. Table 1 also includes simulation results for bursty Poisson arrivals (8-packet bursts with exponential delay between bursts) which indicate negligible effect of burstiness on throughput and n^*.

3. APPROXIMATE SIZING METHOD USING APPLICATION TRAFFIC PROFILE

Our sizing method is applicable to TCP as well as UDP applications. The mapping for UDP applications is simpler since TCP handshake and flow control need not be accounted for. We however describe our method with reference to TCP applications, which can be easily adapted to UDP with minor changes. The method is however not applicable to bulk data TCP applications such as FTP since we do not consider the effect of TCP window control.

We begin by stating the simplifying assumptions: first, that there is no packet loss as perceived by the layers above the MAC. Second, packet arrivals to the MAC are independent of the load on the WLAN. Third, the TCP flows are short lived due to small sized objects being transferred [8].

These assumptions are significant and need justification. Since 802.11 MAC employs MAC-level retransmissions, packet loss is reported to higher layers only if a) the limit on the number of retries is exceeded or b) the interface queue is full. Since we are considering error-free physical layer, the probability of any of these happening is very low until the WLAN has reached saturation. From the simulations, we verified that there were indeed no TCP retransmissions (for error-free channel), even after the application throughput had saturated.

Since TCP connections exhibit closed-loop behavior, the second assumption requires that the MAC delays be constant. Although this is true for lightly loaded WLANs (refer Fig. 1(b)), the effect of this assumption on accuracy near saturation needs to be evaluated.

The third assumption is based on the characterization of HTTP traffic [8] suggesting that 90 % of www objects for static content are smaller than 12 Kbytes. This implies that TCP window size does not alter the offered load to the MAC.

With these assumptions, the overheads of TCP can be approximated to be static for non-saturated WLANs. Our approximate sizing method consists of the following steps:

Step-1) Specification of application workload: Provided either by the user herself or extracted from measurement data, the application workload specification consists of the type of application, the average size of application payload L_{app} (in bytes), the number of users n and their rates of requests $\lambda_{\text{app,i}},\ i \in [1, \cdots, n]$.

Step-2) Characterization of application traffic: Each application request generates a sequence of application messages. In addition, there are control packets at intermediate layers. Analytical models, simulations or empirical evidence can be used for characterizing this stream of packets. The output of this step is the average rate λ_{mac} and average size l_a, of the packets offered to the WLAN interface, averaged over all n stations.

Step-3) Computation of MAC performance metrics using analytical models: The average throughput of the WLAN (in bytes/sec), denoted as S_{mac} is computed from λ_{mac}, l_a and n using an analytical model for 802.11 MAC.

Step-4) Mapping the MAC performance to application performance: Given the MAC throughput S_{mac}, the application throughput S_{app} (in bytes/sec) is obtained, after subtracting overheads (assumed as constant) at TCP and the application layer.

Step-5) Finding the application saturation point: The application saturation point, denoted by n^{**}, is defined as the number of stations for which the application throughput saturates, for a given application request rate λ_{app} per station. Formally, if S_o is the offered load,

$$n^{**} = \text{largest } n \text{ such that } \quad \frac{S_o - S_{\text{app}}}{S_o} < \epsilon_t \qquad (1)$$

where $S_o = L_{\text{app}} \sum_{i=1}^{n} \lambda_{\text{app,i}}$ bytes/sec and ϵ_t is the threshold fractional difference between the offered load and the throughput.

The following algorithm summarizes the computation of n^{**}:

obtain $\lambda_{\text{app}}, L_{\text{app}}$
$n \leftarrow n_{\text{start}}$
repeat
  compute $\lambda_{\text{mac}}, l_a$
  compute S_{mac} using analytical model for the MAC
  compute S_{app}, S_o
  $n^{**} \leftarrow n$, increment n
until $\frac{S_o - S_{\text{app}}}{S_o} < \epsilon_t$

4. SIZING FOR WEB TRAFFIC

We illustrate the above steps with the example of a simple HTTP application. We consider the scenario of a single-hop infrastructure mode WLAN. While uniform traffic from all stations is assumed by the analytical models, their application for the WLAN where downstream traffic is concentrated at the access point is a major approximation which needs to be validated through simulations. In a WLAN of size n, there are $n-1$ HTTP client nodes and one access point. Each HTTP client communicates with a server outside the WLAN through the access point. We do not model the network dynamics outside the BSS for the sake of simplicity.

Step-1) Specification of HTTP workload: The $n-1$ clients send requests to the server for objects of fixed size (L_{app} bytes), the request generation process being Poisson with rate λ_{app}. Thus $\lambda_{\text{app,i}}$ is λ_{app} for all clients and is zero for the access point.

Step-2) Characterization of HTTP traffic: Each HTTP object request generates a GET request of a small size (approximately 40 bytes) and an HTTP response consisting of the payload. The response may be broken into multiple TCP segments. For keeping the analysis simple, we classify packets in two sizes: small packets of size l_h bytes, consisting of just TCP and IP headers, and large packets, of a size approximately equal to the TCP MSS [2] plus l_h.

[2] For both simulations and our method, we use an MSS of 536 bytes, the default value commonly used by many TCP implementations for non-local connections.

For the case of HTTP 1.0 protocol[3] (no keep-alive), for each object request, we observed from packet traces

i) 5 small packets in the upstream direction, and
ii) 4 small and D large packets in the downstream direction,

where $D = \lceil \frac{L_{\text{app}}}{MSS} \rceil$. Then,

$$\lambda_{\text{mac}} = \frac{\lambda_{\text{app}}(n-1)(9+2D)}{n} \quad \text{packets/s/station} \qquad (2)$$

$$\text{Average Packet size} = l_a = \frac{[(9+2D)l_h + L_{\text{app}}]}{(9+2D)} \text{ bytes} \qquad (3)$$

Step-3) Computation of MAC performance metrics: From λ_{mac}, l_a and n, we compute S_{mac} using Foh-Zukerman's model [4].

Step-4) Mapping the MAC performance to the HTTP performance: Since each HTTP request generates $l_h(9+2D) + L_{\text{app}}$ bytes of traffic at the MAC, of every $l_h(9+2D) + L_{\text{app}}$ bytes of MAC throughput, L_{app} bytes constitute application throughput. Thus,

$$S_{\text{app}} = \frac{S_{\text{mac}} L_{\text{app}}}{l_h(9+2D) + L_{\text{app}}} \qquad (4)$$

Step-5) Finding the application saturation point: The value of n satisfying (1) is the application saturation point n^{**}, where the offered application load

$$S_o = (n-1)\lambda_{\text{app}} L_{\text{app}} \qquad (5)$$

5. VALIDATION USING SIMULATIONS

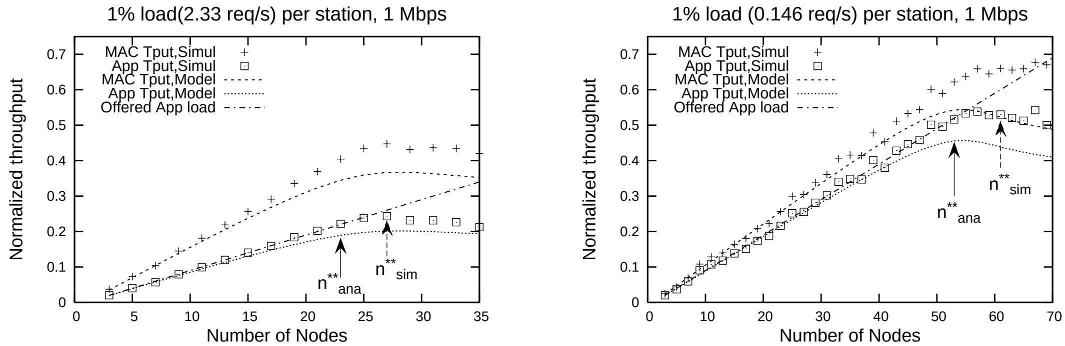
We conducted extensive simulations using the *ns-2* simulator [1] to validate our method. The scenario simulated was identical to that described for the analytical method: $n-1$ stations, each having an HTTP client, sending requests to a server outside BSS through an access point. All wireless stations were within receiving range of each other. We used the `webcache` module to generate HTTP requests according to a Poisson process with rate λ_{app} at each client. The results were obtained from 14 independent sets. In each set, we repeated the simulations (of 30 seconds duration each) for increasing values of n, while keeping λ_{app} and the payload size fixed.

The application and MAC throughput, normalized to the channel bit rate are plotted in Fig. 2(a) and Fig. 2(b) for HTTP object sizes of 536B and 8576B respectively. The plots indicate that the application payload size has a significant impact on the application and MAC throughput. As can be seen from the plots, our sizing method is able to track the variations in application throughput up to saturation [4] reasonably well.

The application saturation points n^{**} for different combinations of channel bit rate, HTTP request rate and HTTP object size are tabulated in Table 2. The per-station load is

[3] Although we have considered HTTP 1.0 only, the method can be extended to HTTP 1.1 by incorporating session information.

[4] Beyond saturation, the application throughput for simulations drops due to closed-loop behavior of TCP while that for the analytical method drops due to MAC saturation. We omit further details due to space limitations.



(a) 536B objects, 1Mbps

(b) 8576B objects, 1Mbps

Figure 2: Application and MAC throughput vs n

Req.Rate (req/stn/s)	% Load /stn	BitRate (Mbps)	Payload (bytes)	$n^{**}_{\rm sim}$	$n^{**}_{\rm ana}$
2.332	1.0	1	536	27	23
0.292	1.0	1	4288	59	49
0.146	1.0	1	8576	61	53
4.664	2.0	1	536	15	10
4.664	0.182	11	536	30	29
0.583	2.0	1	4288	29	25
0.583	0.182	11	4288	87	93
0.292	2.0	1	8576	31	27
0.292	0.182	11	8576	99	107
6.996	3.0	1	536	10	5
6.996	0.273	11	536	18	17
0.875	3.0	1	4288	20	16
0.875	0.273	11	4288	60	63
0.437	3.0	1	8576	21	17
0.437	0.273	11	8576	66	73

Table 2: Saturation Points n^{**}, $\epsilon_t = 0.15$

the application payload per unit time, expressed as a percentage of the channel bit rate. The $n^{**}_{\rm sim}$ and $n^{**}_{\rm ana}$ values are n^{**} values obtained from simulations and from our method respectively, for a ϵ_t value of 0.15 . We found the prediction of n^{**} to be sensitive to the choice of ϵ_t. The throughput and n^{**} values at 11Mbps [5] do not scale up proportional to the data rate due to physical layer overheads which become dominant because of large number of small control packets. Despite the approximation of symmetric traffic distribution across all wireless stations and our simplifying assumption of load-independent packet arrivals, the n^{**} values suggested by our method are within 13% of those derived from simulations for a majority of the parameter sets.

6. CONCLUSION

We have proposed a sizing method for recommending the maximum number of users in a 802.11 WLAN. A major contribution of our method is that it derives, through approximations, a mapping from the application-layer workload specification to the link-layer traffic, for TCP applications over WLANs which are difficult to model. This mapping facilitates the usage of existing analytical models for 802.11 MAC in sizing.

We have illustrated our method using an example of HTTP traffic. A close match between the results from our method and *ns-2* simulations, particularly the value of n^{**}, suggests that our method works quite well for the scenario considered despite the approximations. The computation of n^{**} can be made more accurate and less sensitive to the choice of ϵ_t by modeling the closed-loop behavior of TCP.

The simple example described in this paper is for validating the method and its assumptions. A more realistic application profile, rather a mix of applications needs to be characterized for practical use. The method can be extended for a combination of different applications by aggregating the packet streams they generate. Application-layer delays, although more difficult to obtain, can also be derived using similar technique for delay-constrained sizing.

[5] We did not use the "short preamble" option of 802.11b for both simulations and analytical method.

7. REFERENCES

[1] The network simulator ns-2. http://www.isi.edu/nsnam/ns/.

[2] G. Bai and C. Williamson. Simulation evaluation of wireless web performance in an IEEE 802.11b classroom area network. In *Proc. of LCN '03*, pages 663–672, October 2003.

[3] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE JSAC*, 18(3):535–547, March 2000.

[4] C. Foh and M. Zukerman. Performance analysis of IEEE 802.11 MAC protocol. In *Proc. of the European Wireless Conference*, pages 184–190, February 2002.

[5] D. Kotz and K. Essien. Analysis of a campus-wide wireless network. In *Proc. of MOBICOM '04*, pages 107–118, September 2002.

[6] R. Litjens, F. Roijers, J. van den Berg, R. Boucherie, and M. Fleuren. Performance analysis of wireless LANs: An integrated packet/flow level approach. In *Proc. of 18th International Teletraffic Congress*, 2003.

[7] Y. C. Tay and K. C. Chua. A capacity analysis for the IEEE 802.11 MAC protocol. *Wireless Networks*, 7(2):159–171, 2001.

[8] Z. Zhu, Y. Mao, , and W. Shi. Workload characterization of uncacheable HTTP traffic. Technical Report MIST-TR-03-003, MIST, Dept. of Computer Science, Wayne State University.