ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 1, Issue 4, October 2012

An Enhanced Model for Achieving High Throughput using Variant Buffer Size Methodology

RSVV. Prasad Rao[#1], M.V. Rajesh[*2]

[#]*PG Scholar,*
Aditya Engineering College, Surampalem, AP, India.
[1*]*Assistant Professor, Aditya engineering college, SURAMPALEM, AP, India*

***Abstract*— This paper is an attempt to propose an enhanced model for achieving high throughput in wireless networks especially in 802.11 using variant buffer size methodology. In this paper we consider sizing of buffers in wireless networks. Buffers are temporary storage areas which are needed for holding the packets that travel from source station to destination station on the network. So deciding the size of buffer to which value is always a major concern, especially in wireless networks. We consider this as a problem as one cannot decide how many stations are active on a wireless network at a given point of time and so it is difficult to estimate the amount of network traffic on such a wireless network. As it is difficult to estimate the number of active wireless stations on a network it is difficult to predict the average number of packets that are traveling on the network that are to be routed through the router present between the two ends of the communication stations. So we cannot come to a conclusion on the size of buffer if the traffic is variable on the wireless network. If we wish to set the size of buffer to some constant value then there are problems such as reduced throughput, degraded performance because if the size of buffer is set to some constant value then the size may not be sufficient which leads to drop outs in the packets due to lack of buffer space. In case if we wish to set the buffer size to some high value then the problem is underutilization in case the amount of packets that are traveling are below the set buffer size threshold. So deciding the buffer size in both of the above cases is always a big question. Generally the buffer size is determined taking the product of BANDWIDTH and DELAY into consideration which is called as BDP. This BDP algorithm suggests taking the buffer size to be constant which leads to poor performance. So in this paper we propose a system using a dynamic algorithm called eBDP (Emulating Bandwidth Delay Product) algorithm to decide the size of buffer that dynamically adjusts the size of buffer depending on the network traffic requirements.**

***Keywords*— Buffer sizing, eBDP, wireless networks, variant buffer sizing, MAC**

I. INTRODUCTION

The distribution of packet service times is also strongly dependent on the WLAN offered load. This directly affects the burstiness of transmissions and so buffering requirements. IEEE 802.11b support up to 11 Mbps, sometimes this is not enough – far lower than 100 Mbps fast Ethernet In co-existing environment, the probability of frequency collision for one 802.11 frame vary from 48%-62%. The following are Disadvantages of existing model: 1) Unaware of interference from/to other networks 2) Weak security policy 3) Poor performance (coverage, throughput, capacity, security) 4) Unstable service 5)Customer dissatisfaction

The sizing of buffers in wireless networks (especially those based on 802.11/802.11e) appears to have received very little attention within the networking community. Exceptions include the recent work relating to buffer sizing for voice traffic in 802.11e WLANs, work which considers the impact of buffer sizing on TCP upload/download fairness, and work in which is related to 802.11e parameter settings. Buffers play a key role in 802.11/802.11e wireless networks. To illustrate this, we present measurements from the production WLAN of the Hamilton Institute, which show that the current state of the art which makes use of fixed size buffers, can easily lead to poor performance. We recorded RTTs before and after one wireless station started to download a 37MByte file from a web-site. Before starting the download, we pinged the access point (AP) from a laptop 5 times, each time sending 100 ping packets. The RTTs reported by the ping program was between 2.6-3.2 ms. However, after starting the download and allowing it to continue for a while, the RTTs to the AP hugely increased to 2900-3400 ms. This reduction in delay does not come at the cost of reduced throughput, i.e., the measured throughput with the A* algorithm and the default buffers is similar. In this paper, we consider the sizing of buffers in 802.11 /802.11e based WLANs. We focus on single hop WLANs since these are rapidly becoming ubiquitous as the last hop on home and office networks as well as in so called "hot spots" in airports and hotels, but note that the proposed schemes can be easily applied in multi-hop wireless networks. Our main focus in this paper is on TCP traffic since this continues to constitute the bulk of traffic in modern networks (80–90% of current Internet traffic and also of WLAN traffic), although we extend consideration to UDP traffic at various points. We propose the following advantages in this model: a)The reduction in network delay not only benefits UDP traffic, but also short-lived TCP connections b)Comes from easy maintenance, cabling cost, working efficiency and accuracy c)Network can be established in a new location ust by moving the PCs.

ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 1, Issue 4, October 2012

II. Related Works

Prasad & et.al [1] proposed a technique of Router buffer sizing for TCP traffic and its role in output or input capacity ration. The main objective of the paper is to find the buffer size that maximizes the average per-flow TCP throughput. They [1] considered two things for achieving the objective: First, they considered the more realistic case of non-persistent TCP flows with heavy-tailed size distribution. Second, instead of only looking at link metrics, they considered to focus on the impact of buffer sizing on TCP performance.

Mihaela Enachescu & et.al [2] initiated a technique of Routers with very small buffers. They explored how buffers in the backbone can be significantly reduced even more, to as little as a few dozen packets. They argued that if the TCP sources are not overly bursty, then fewer than twenty packet buffers were sufficient for high throughput. Specifically, they argued that O(log W) buffers are sufficient, where W is the window size of each flow. They supported their claim with analysis and a variety of simulations. The change they need to make to TCP is minimal which each sender just needs.

Tianji Li & et. Al [3] proposed an adaptive buffer sizing technique for TCP flows in 802.11 e WLANs. They considered the provision of access point buffers in WLANs. They first demonstrated that the default use of static buffers in WLANs leads to either undesirable channel under-utilisation or unnecessary high delays, which motivates the use of dynamic buffer sizing. Although adaptive algorithms have been proposed for wired Internet, a number of fundamental new issues arise in WLANs which necessitates new algorithms to be designed. These new issues include the fact that channel bandwidth is time-varying, the mean service rate was dependent on the level of channel contention, and packet inter-service times vary stochastically due to the random nature of CSMA/CA operation. They proposed an adaptive sizing algorithms which is demonstrated to be able to maintain high throughput efficiency whilst achieving low delay.

Guido Appenzeller & et. al [4] proposed sizing router buffers. They [4] argued that the rule-of-thumb (*B* = (overline*RTT* x *C*) was then outdated and incorrect for backbone routers. This was because of the large number of flows (TCP connections) multiplexed together on a single backbone link. Using theory, simulation and experiments on a network of real routers, they [4] showed that a link with *n* flows requires no more than *B* = (overline*RTT* x *C*) √*n*, for long-lived or short-lived TCP flows. The consequences on router design are enormous: A 2.5Gb/s link carrying 10,000 flows could reduce its buffers by 99% with negligible difference in throughput; and a 10Gb/s link carrying 50,000 flows required only 10Mbits of buffering, which can easily be implemented using fast, on-chip SRAM.

Malone & et.al [6] presented their technique on buffer sizing for voice in 802.11 WLANs. The use of 802.11 to transport delay sensitive traffic was becoming increasingly common. This raised the question of the trade off between buffering delay and loss in 802.11 networks. They [6] found that there existed a sharp transition from the low-loss, low-delay regime to high-loss, high-delay operation. Given modest buffering at the access point, that transition determined the voice capacity of a WLAN and its location was largely insensitive to the buffer size used.

III. Proposed Technique

In 802.11 networks, at the wireless station the mean service time and the distribution time of service is varying. The variation in the mean service time and the distribution time of service is varying because of the few reasons like variation in the number of active wireless stations and the variations in the loads on the WLAN and changes in the physical transmit rate used (i.e. in response to changing radio channel conditions).

We begin by considering a simple adaptive algorithm based on the classical BDP rule. Although this algorithm cannot take advantage of statistical multiplexing opportunities, it is of interest both for its simplicity and because it will play a role in the more sophisticated A* algorithm developed in the next section. As noted previously, and in contrast to wired networks, in 802.11 WLANs the mean service time is generally time varying (dependent on WLAN load and the physical transmit rate selected by a station) consequently, there does not exist a fixed BDP value.

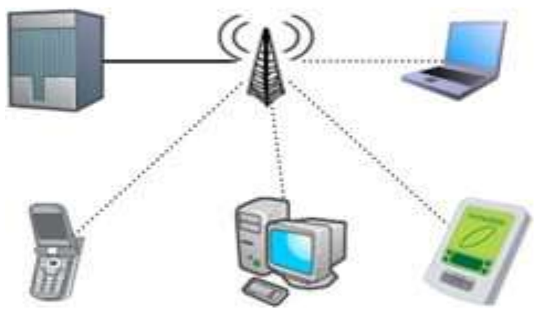

Figure 1:Proposed Architecture

However, we note that a wireless station can measure its own packet service times by direct observation, i.e., by recording the time between a packet arriving at the head of the network interface queue ts and being successfully transmitted te (which is indicated by receiving correctly the corresponding MAC ACK). Note that this measurement can be readily implemented in real devices, e.g. by asking the hardware to raise an interrupt on receipt of a MAC ACK, and incurs only a minor computational burden. Averaging these per packet service times yields the mean service time Tserv. To accommodate the time-varying nature of the mean service time, this average can be taken over a sliding window. In this paper, we consider the use of exponential smoothing So, to isolate signal discontinuities one would like to have some short basis functions. In order to obtain detailed frequency analysis, one would like to have some very long basis functions. i.e. wavelet analysis provides immediate access to information that can be obscured by other time frequency methods such as Fourier analysis. Wavelet analysis has begun to play including signal processing, data and image compression, solution for partial differential equations, modeling multiscale phenomena and statistics etc. It provides a systematic way to represent and analyze multiscale structure. It also provides a systematic and universal representation for wide classes of functions. Multiscale representation is a representation selected for the convenience of the analyst, because it often provides an efficient representation of information for storage, calculation or communication of information. Tserv(k + 1) = (1 − W)Tserv(k) + W(te − ts) to calculate a running average since this has the merit of simplicity and statistical robustness (by central limit arguments).

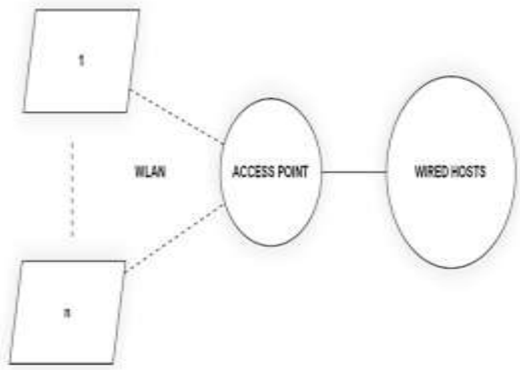

Figure 2: WLAN Topology used in Simulations

The choice of smoothing parameter involves a trade-off between accommodating time variations and ensuring the accuracy of the estimate – this choice is considered in detail later. Given an online measurement of the mean service time Tserv, the classical BDP rule yields the following eBDP buffer sizing strategy. Let Tmax be the target maximum queuing delay. Noting that 1/Tserv is the mean service rate, we select buffer size QeBDP according to QeBDP =min(Tmax/Tserv,QeBDP max) where QeBDP max is the upper limit on buffer size. This effectively regulates the buffer size to equal the current mean BDP. The buffer size decreases when the service rate falls and increases when the service rate rises, so as to maintain an approximately constant queuing delay of Tmax seconds. We may measure the flows' RTTs to derive the value for Tmax in a similar way to measuring the mean service rate, but in the examples presented here we simply use a fixed value of 200ms since this is an approximate upper bound on the RTT of the majority of the current Internet flows.

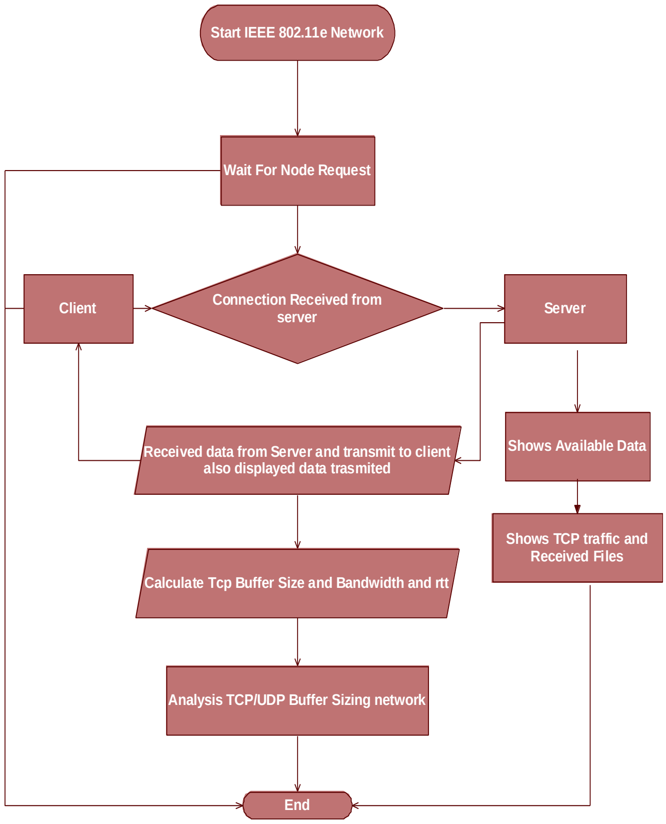

Fig. 1 Flow of our proposed methodology

A. MAC OPERATION OF eBDP algorithm

1: Set the averaging parameter W.
2: **for** each outgoing packet *p* **do**
3: Record service start time t_s for p.
4: Wait until receive MAC ACK for p, record service end time te.
5: Calculate service time of p: $T_{serv} = (1 - W)T_{serv} + W(t_e - t_s)$.
6: **end for**

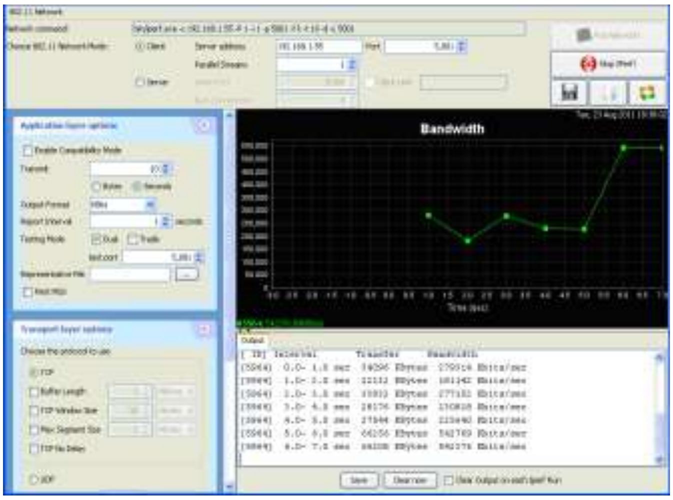

Fig. 4 Experimental Result

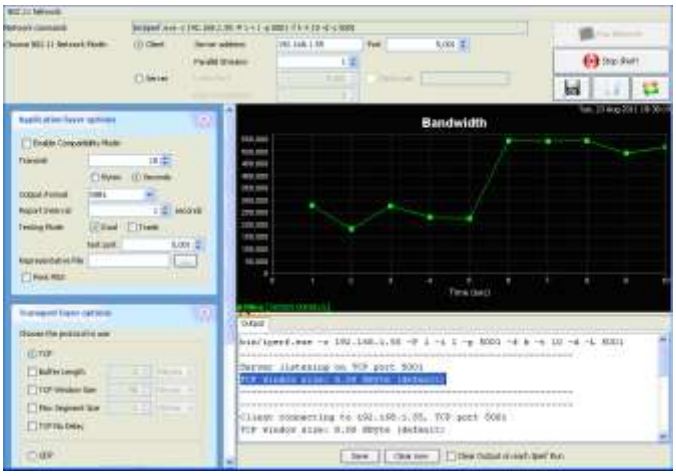

Fig. 5: Performance of the eBDP algorithm as the number of upload flows

IV. CONCLUSIONS

We consider the task of sizing buffers for TCP flows in 802.11e WLANs. A number of fundamental new issues arise compared to wired networks, including the fact that the mean service rate is dependent on the level of channel contention and packet inter-service times vary stochastically due to the random nature of CSMA/CA operation. Motivated by these observations we propose an adaptive buffer sizing algorithm which emulates the classical BDP rule and demonstrate its efficacy via simulations. So in this paper we proposed a system using a dynamic algorithm called eBDP (Emulating Bandwidth Delay Product) algorithm to decide the size of buffer that dynamically adjusts the size of buffer depending on the network traffic requirements. Buffer sizing while rate adaptation is enabled is left as future work, although we believe that the proposed algorithm will work. Future work also includes consideration of the possibility of reducing buffer sizes when multiplexing occurs.

V. REFERENCES

[1] R. Prasad, C. Dovrolis, and M. Thottan, "Router Buffer Sizing for TCP Traffic and the Role of the Output/Input Capacity Ratio ," *IEEE/ACM Transactions on Networking*, 2009

[2] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden,"Routers with Very Small Buffers," in Proc. Of INFOCOM, Dec. 2006.

[3] T. Li and D. Leith, "Adaptive Buffer Sizing for TCP Flows in 802.11e WLANs," Chinacom 2008.

[4] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," *Proc. ACM SIGCOMM*, 2004.

[5] C. Chatfield, *The Analysis of Time Series, An Introduction*, CRC Press 2004.

[6] D. J. Leith, P. Clifford, D. Malone, and A. Ng, "TCP Fairness in 802.11e WLANs," *IEEE Communications Letters*, vol. 9, no. 11, pp 964–966, 2005.

[7] D. Malone, P. Clifford, and D. J. Leith, "On Buffer Sizing for Voice in 802.11 WLANs," *IEEE Communications Letters*, vol. 10, no. 10, pp 701–703, 2006.

[8] S. Pilosof, et. al., "Understanding TCP fairness over Wireless LAN," in *Proc. IEEE Infocom* 2003.

[9] R. Stanojevic, C. M. Kellett, and R. N. Shorten, "Adaptive Tuning of Drop-Tail Buffers for Reducing Queueing Delays," *IEEE Communications Letters*, vol. 10, no. 7, pp 570–572, 2006.

[10] M. Thottan, and M. C. Weigle, "Impact of 802.11e EDCA on mixed TCP-based applications," in *Proc. IEEE WICON* 2006.

[11] Z. Zhao, S. Darbha, and A. L. N. Reddy, "A method for estimating the proportion of nonresponsive traffic at a router," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 708–718, 2004.