Optimizing the Channel Load Reporting Process in IEEE 802.11k-enabled WLANs

Emmanuel A. Panaousis, Pantelis A. Frangoudis, Christopher N. Ververidis, George C. Polyzos
Mobile Multimedia Laboratory, Department of Computer Science
Athens University of Economics and Business
Athens 104 34, Greece
{panaousis06, pfrag, chris, polyzos}@aueb.gr

***Abstract*—IEEE 802.11k is an extension of the IEEE 802.11 specification for radio resource measurements. In an IEEE 802.11k-enabled wireless LAN, an access point or other network element may request from a client or another access point to monitor and report the load of a channel. We call the latter a channel monitoring station. In this paper we propose a mechanism for a channel monitoring station to efficiently derive accurate values of channel load. We especially focus on optimizing the duration of channel monitoring and thus minimize the impact on applications. Note that such mechanisms are critical for the success of new sharing regimes such as *Cognitive Radio* and *Open Spectrum Access*.**

I. Introduction

IEEE 802.11 Task Group k develops an extension to IEEE 802.11 wireless local area network (WLAN) specification for radio resource measurements. According to this extension a radio station can measure and assess the radio environment and take corresponding actions. IEEE 802.11k [1], [2] specifies types of radio resource information to measure and the associated request and report mechanisms and frame formats through which the measurement requests and results are communicated among stations. The extension defines different types of measurements which provide information to discover the best available access point.

IEEE 802.11k is a proposed standard describing how a wireless local area network should perform channel selection, roaming, and power control in order to optimize network performance. It is intended to improve the way traffic is distributed within a network. In a WLAN, each device normally connects to the access point which provides the strongest signal. The arrangement of the users to the access points can sometimes lead to excessive demand on one access point and underutilization of others. This results in degradation of overall network performance. In a network conforming to IEEE 802.11k, if the access point having the strongest signal is loaded to its full capacity, a wireless client is connected to one of the access points with lower utilization. Despite the weaker signal for this client, the overall throughput is greater because more efficient use is made of the wireless network resources.

In this paper, we study the channel load measurement for IEEE 802.11k. An access point or a WLAN switch can request from a client or another access point data about channel load or how long the channel was used during a given time. A client, or some other access point, called a *channel monitoring station*, monitors one or more channels and collects load information for a period called *channel monitoring duration*.

After the channel monitoring process[1], the results are reported to the requesting entity through a *channel load report*, as depicted in Fig. 1. Specifically, the channel monitoring station reports the fractional duration of the period over which the physical or virtual carrier sense mechanism indicates that the medium is busy. As a result, the station that issued the request is informed about the load of a channel. In other words this parameter gives an idea of how many free slots a new station would have at its disposal. When another request is received, the channel monitoring station repeats the measurement and reports the results in the same way. An example of a virtual carrier sensing mechanism is the *Clear Channel Assessment* (CCA) [3]. It is a logical function found within physical layers which determines the current state of usage of a wireless medium. Such a function is found in IEEE 802.11 networks and helps with contention avoidance.

We focus on the measurement process for channel load (utilization). Specifically, a channel monitoring station is sampling the wireless channel access pattern and computes the confidence interval and the estimated mean value of the load. We propose a mechanism that optimizes the channel monitoring duration in order to find an estimate of the true channel load.

The remainder of the paper is organized as follows. In Section II we examine related work and we discuss fundamental concepts. In Section III we describe the proposed methodology and an algorithm for implementing the channel monitoring process, while in Section IV we present simulation results. We present our conclusions and plans for future work in Section V.

II. Related Work

The work presented in [4] serves as the basis for our study. In that paper the authors propose a method to estimate the confidence of the channel load measurement results. For that purpose, they apply the concept of confidence intervals to IEEE 802.11k radio resource measurements, as applied to stochastic processes. Our main contribution consists of the following extensions to their work:

[1] Also referred to as the channel measuring process or channel sensing process.

Proceedings of the 2008 16th IEEE Workshop on Local and Metropolitan Area Networks
978-1-4244-2028-5/08/$25.00 ©2008 IEEE

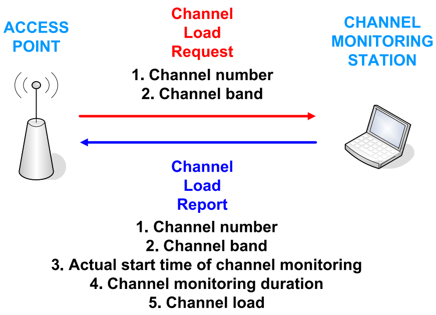

Fig. 1. An access point requests a channel load report from a client which then is called a *channel monitoring station.*

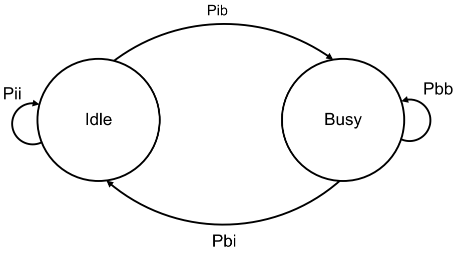

Fig. 2. The Gilbert model and the corresponding probabilities.

- We propose that the channel monitoring process is performed in *channel monitoring sub-periods*, so that the overhead of re-calculating the confidence interval of the measured mean channel load is reduced, since confidence intervals are re-calculated only at the end of each sub-period and not each time a new sample is acquired.
- We present a simple algorithm for performing the channel monitoring process based on the concept of confidence intervals and we introduce some termination criteria which can help speed up the measurement process, while still reporting accurate results.

A. The Gilbert Model

The Gilbert-Elliot model, or simply the Gilbert model, is one of the simplest to consider for modeling channel state [5], [6]. To be specific, we assume that the idle and busy states of a channel are modeled as depicted in Fig. 2. The idle and busy states have stationary probabilities p_i and p_b, respectively.

We denote the transition probabilities by, P_{xy}, as shown in Fig. 2.

The Markov model framework allows us to determine how much time the system spends in each state and the probability of being in a particular state. We assume discrete Markov chains with the duration of the slot equal to $0.02\ ms$ which is the slot duration in IEEE 802.11b.

The number of slots a channel stays in one state is geometrically distributed.

For the stationary probabilities we have:

$$p_b = \frac{P_{ib}}{P_{ib} + P_{bi}},\ p_i = \frac{P_{bi}}{P_{bi} + P_{ib}} \tag{1}$$

B. Confidence Intervals

A confidence interval [7] is an interval estimate of a population parameter. In order to avoid to estimate the parameter by a single value, an interval which it includes with probability ϱ is given. Increasing the desired ϱ will widen the confidence interval.

Suppose that n identically distributed samples $X_1, X_2, ..., X_n$ have been generated by repeated trials of some experiment. Let μ and $\sigma > 0$ be the mean and standard deviation, respectively, of X_k. The mean μ is taken to be an unknown quantity that is to be estimated from the samples of X_k. The standard deviation σ is given by the equation:

$$\sigma = \sqrt{(\sum_{i=1}^{n}(X_i - \mu)^2)/(n-1)} \tag{2}$$

According to Student's t-distribution [8] tables, with $n-1$ degrees of freedom and confidence level ϱ, we compute the value of t. Finally, we compute the confidence interval CI of these sample, as:

$$\mu - \frac{t \cdot \sigma}{\sqrt{n}} \leq CI \leq \mu + \frac{t \cdot \sigma}{\sqrt{n}} \tag{3}$$

A confidence interval is useful only if the underlying statistics are unbiased [4], which is usually the case in radio resource measurements.

III. Proposed Methodology

As we have discussed, according to IEEE 802.11k, a channel monitoring station is asked to report channel load information to an access point or a WLAN switch. The latter uses this information to select a "good" channel to operate. Its goal is to achieve as high throughput as possible.

In this paper, we simulate the wireless channel utilization using the Gilbert model as described in the previous section. To estimate Gilbert model parameters (i.e. transition probabilities) for realistic WLAN settings, we have carried out a set of simulations using *ns2*[2], as described in Section IV-B.

The channel pattern can be described by a series of 0 (for idle) and 1 (for busy) values taken during a period equal to the *channel monitoring duration*.

A. Channel monitoring sub-period

During a *channel monitoring sub-period*, a channel monitoring station is taking samples from the channel. The sampling period of taking these samples is determined by the *quality* of channel monitoring station. As we will discuss in Section IV-A, high-quality monitoring stations are capable of collecting samples with a higher rate.

[2] http://www.isi.edu/nsnam/ns/

After the expiration of this period, the station stops sampling and:

- Computes the confidence interval and the estimated mean value of the set of samples taken during the current and all the previous channel monitoring sub-periods.
- Computes the width of this confidence interval, let it be $w_{current}$.
- Computes the *improvement ratio* value, let it be

  $$(w_{previous} - w_{current})/w_{previous}$$

  where $w_{previous}$ is the width of the confidence interval computed from the samples of all the previous channel monitoring sub-periods, only.

The confidence interval will be an interval of values, let it be $[\alpha, \beta],\ 0 \leq \alpha,\ \beta \leq 1$. If the channel monitoring process is accurate the true channel load will be within this interval with a probability equal to the confidence level, as discussed in Section II-B.

After the first channel monitoring sub-period and the computation of the first confidence interval the station is repeating the same process for another one channel monitoring sub-period, at least. The station keeps monitoring until it computes an "appropriate" confidence interval. This methodology has emerged from the need of having as more efficient, accurate and reliable estimation of the channel load as possible.

But when does this iterative procedure stop? In other words which confidence interval is considered "appropriate"? In our proposed mechanism for channel load monitoring, which is presented in Section III-B, we assume two termination criteria.

B. The algorithm

Every channel monitoring station implements the process described in Algorithm 1. This includes both the steps of channel monitoring and also the computation of the confidence intervals and of the estimated mean load values.

Considering line 5, note that in the calculation of each confidence interval and of each estimated mean value we take into account the samples taken during all the previous channel monitoring sub-periods in addition to the samples taken during the current channel monitoring sub-period.

The quality of a channel monitoring station system determines the value of r (line 4). This parameter, as well as the critical values of $w_{default}$, $improvement_{min}$ and channel monitoring sub-period, which optimize the performance of channel monitoring process, will be discussed in Section IV.

As to the improvement ratio (line 6), it should be noted that it is only considered in case $w_{previous} \geq w_{current}$. Otherwise, a negative improvement ratio value could cause the algorithm to terminate prematurely.

C. Reporting

When the channel monitoring process stops, the station reports the results to the requesting entity. As a result, the latter has an estimation of channel occupancy with a confidence level equal to a preselected value.[3] Specifically, the reporting results include the following information:

- Channel number: the number of monitored channel in the existing IEEE 802.11k-enabled WLAN.
- Channel band: the band of frequencies of the monitored channel.
- Actual start time of monitoring: the starting time of the channel monitoring process.
- Estimate of channel load: the final computed confidence interval of channel load and the final estimated channel mean load.
- Channel monitoring duration: is the total time of channel monitoring process.

Algorithm 1 Channel load measurement process

```
1:  if a channel load report is requested then
2:    while monitoring the channel do
3:      for the next period equal to channel monitoring sub-period do
4:        sample the channel with rate $r$
5:        calculate the confidence interval and the estimated mean value of all the taken samples
6:        if the width of last computed confidence interval, $w_{current}$, is smaller than a default value, let it be $w_{default}$ or the improvement ratio value is smaller than a minimum ratio, let it be $improvement_{min}$ then
7:          terminate monitoring process
8:          report the results of channel monitoring process to the requesting entity
9:          wait for a new reporting request
10:       else
11:         return to line 3
12:       end if
13:     end for
14:   end while
15: end if
```

IV. Simulation Results

In this section, we provide results of the performance assessment of the proposed mechanism. First, we need to define suitable metrics. We have used Matlab^{TM}and *ns2* to implement our simulation environment.

A. Channel monitoring station parameters

Similarly to [4] we assume two types of channel monitoring stations with respect to the accuracy level and their monitoring capabilities. These categories are purely hypothetical as there are no such systems out in the market and their development may be a proposal for the future. Specifically, we assume:

1) *High cost/high quality* channel monitoring stations: these stations can be deployed in public access networks, for the purpose of monitoring. They sample at a higher

[3] We provide results with confidence levels 95% and 99%.

rate and report more accurate confidence intervals and estimated mean load values than low quality channel monitoring stations.

2) *Low cost/low quality* channel monitoring stations: these systems monitor with reduced effort. They report results with less confidence and may be used in low-end systems such as IEEE 802.11k-enabled PDA Wi-Fi adapters.

Each type of radio channel monitoring station determines a value of r^4 and a value of $w_{default}{}^5$. Obviously, a high quality channel monitoring station sets its sampling period smaller than a low quality radio system in order to be more accurate. Namely, the high quality channel monitoring station exerts more effort taking more samples from the channel pattern. In the following, we discuss the parameters of r and $w_{default}$ considered in our simulations, with respect to the quality level of the channel monitoring station:

- *High quality* channel monitoring station parameters: we assume that a high quality channel monitoring station samples every $\frac{1}{r} = 2ms$.
- *Low quality* channel monitoring station parameters: we assume that a low quality channel monitoring station samples every $\frac{1}{r} = 4ms$.

It's critical here to mention that, according to Mangold and Berleman [4], we should not sample at a very high frequency (for example, once every 0.5ms). In their paper, they have shown that picking a high sampling rate should be done with care to avoid *oversampling*. They have shown that sampling with a higher rate, although reducing confidence intervals, may lead to invalid conclusions as to the mean load and does not necessarily improve accuracy.

B. Gilbert model parameters

Since we chose the Gilbert model to represent channel access patterns, we needed a means of selecting appropriate model parameters (i.e. transition probability values). To get an insight on what these values are for realistic WLAN settings, we performed a set of simulations.

In particular, we considered a WLAN cell where wireless clients are attached to an AP connected to a wired host. These wireless clients executed applications like FTP or VoIP or their combination. Measuring the number of busy and idle slots (slot duration was set to $20\mu sec$, according to IEEE 802.11b) and the number of transitions between channel states, we estimated the Gilbert model's parameters, as well as channel load.

We simulated three application usage cases, as follows:

- Each wireless node transfers data using FTP to the wired host, via the AP.
- Each wireless client sets up a VoIP session with the wired host. VoIP traffic is bidirectional and we have assumed that the G.729 voice coded is used, without silence suppression. This generates CBR traffic, carried over UDP. Every VoIP packet has 32 bytes of payload (20 bytes of audio and 12 bytes for the RTP header) and 50 packets per second are sent for each call direction.
- Each wireless client simultaneously executes both the FTP and VoIP applications, as described above.

TABLE I
GILBERT PARAMETERS FOR FTP TRAFFIC

Number of nodes	P_b	P_i	P_{ib}	P_{bi}
1	0.795	0.205	0.103	0.027
5	0.805	0.195	0.091	0.022
15	0.814	0.186	0.094	0.021
25	0.815	0.185	0.094	0.021

TABLE II
GILBERT PARAMETERS FOR VOIP TRAFFIC

Number of nodes	P_b	P_i	P_{ib}	P_{bi}
1	0.364	0.636	0.021	0.036
5	0.841	0.159	0.160	0.030
15	0.873	0.127	0.197	0.029
25	0.882	0.118	0.212	0.028

For each of the above application cases, we carried out experiments where the number of wireless clients was 1, 5, 15 and 25. In these simulations, which were implemented using *ns2*, IEEE 802.11b at 11Mbps was used, with the RTS/CTS option disabled. All wireless nodes and the AP were within one another's transmission range. The AP and the wired host were connected using 100Mbps Ethernet. Each simulation was run for 5 minutes (simulation time).

The results from the above simulations are shown in Tables I, II and III. For each case, we report the probabilities that the channel is busy or idle and the transition probabilities from the busy to the idle states and vice versa. We then use these values to generate channel access patterns (according to the Gilbert model), to which we apply and evaluate our channel monitoring algorithm.

C. Default width parameter ($w_{default}$)

For both *high* and *low* quality monitoring stations, we use the following algorithm termination criteria. For confidence level equal to 95% we set $w_{default} = 0.1$. With this value as an assumption we found that the value of $w_{default}$, in

TABLE III
GILBERT PARAMETERS FOR FTP AND VOIP TRAFFIC

Number of nodes	P_b	P_i	P_{ib}	P_{bi}
1	0.782	0.218	0.112	0.031
5	0.841	0.159	0.159	0.030
15	0.873	0.127	0.198	0.029
25	0.883	0.117	0.213	0.028

[4] See line 4 in our algorithm.

[5] See line 6 in our algorithm.

which the *channel monitoring duration* for confidence level 99% holds approximately as in the case of 95%, is 0.15. Thus, a channel monitoring station reports a channel load with ±5% accuracy with confidence level 95% and a confidence interval with ±7.5% accuracy with confidence level 99%.

D. Improvement ratio parameter ($improvement_{min}$)

In addition to the $w_{default}$ value, we need to set a constraint for the *improvement ratio*[6], in our simulations. After experimentation, we have observed that for a value less than $improvement_{min} = 0.03$, channel monitoring duration is overly large. For $improvement_{min} = 0.03, 0.04, 0.05$, we observed that the channel monitoring duration is smaller and the resulting confidence interval width is tolerable.

To summarize, we fix the value of $improvement_{min}$ to 0.03. Thus, we consider that at the point where the change in confidence intervals' width is less or equal to 0.03, namely 3%, the channel monitoring process stops. This criterion helps avoid increased measurement delays without significant decrease in the confidence interval width.

E. Channel monitoring sub-period

As we have discussed[7], the process of sampling holds, each time, for a period called *channel monitoring sub-period*. Thereupon, the station stops sampling and calculates the confidence interval of the set of samples and the estimated mean load. In the calculation of each confidence interval and of each estimated mean load, we take into account the samples taken during all the previous channel monitoring sub-periods in addition to the samples taken during the current channel monitoring sub-period.

In our simulations, we have fixed the channel monitoring sub-period to $20msec$. We selected this value after a series of simulations, where we observed that, for a channel monitoring sub-period greater than $20ms$, the *channel monitoring duration* of some stations is significantly increased. In this case, the response time of the reporting process would also increase. The tradeoff that emerges is between performing excessive measurement using short periods, reducing response time, and frequent recomputation of the confidence interval.

One might claim that the applied channel monitoring duration, although optimized compared to monitoring periods shown in [4], is still too much for real-time applications to tolerate. While this is reasonable to assume at first glance, it should be noticed that the effects of this process on user-perceived application performance highly depend on the frequency of channel load measurement requests. If a station is requested to perform channel load measurements with "reasonable" frequency, e.g. few times per minute, considering that radical changes in spectrum access usage, such as a new AP or client station appearing at a certain area, may not happen very often, it is not obvious that the performance of real-time applications like VoIP will significantly be impacted.

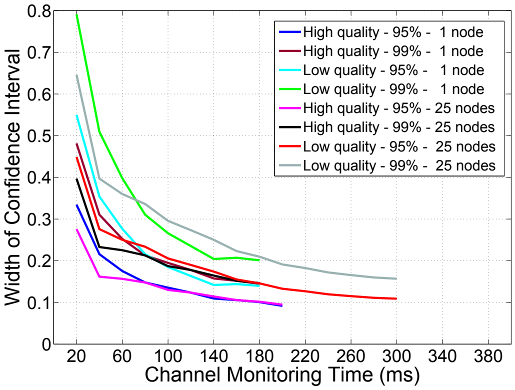

Fig. 3. CI width as a function of the channel monitoring time. The channel corresponds to the case when all wireless nodes are running FTP.

F. Mechanism evaluation

We have implemented our channel monitoring algorithm in Matlab and simulated it on various channel access patterns that we derived using the Gilbert model, with model parameters that we calculated in Section IV-B.

Figures 3, 4 and 5 show the width of the computed confidence intervals as a function of the channel monitoring duration, calculated by a low and a high quality channel monitoring station. We have assumed two confidence levels, 95% and 99%. Due to space limitations, we have only included the results for channels corresponding to 1 and 25 wireless nodes (see Section IV-B).

We focus on examining the change in the width of the computed confidence intervals as a function of the channel monitoring duration. We have observed that the width of confidence intervals is decreasing when channel monitoring duration is increasing, as we expected. Also, typically, for low quality stations (lower sampling rate), the algorithm takes more time to estimate channel load accurately.

Tables IV, V and VI present the confidence intervals, mean sampled load and total measurement duration for the scenarios that we have simulated. In all cases, the calculated confidence intervals included the true channel load and the monitoring process terminated in a reasonable time (less than $300msec$).

V. Conclusion

In this paper, we investigated the channel load report mechanism of the IEEE 802.11k standard. In particular, we proposed a technique, based on confidence intervals, to monitor a wireless channel and report an accurate estimate of the channel's load with the minimum monitoring cost. We derived conditions so that the *channel monitoring duration* is minimized and confidence intervals calculation overhead is reduced. We used the Gilbert model to represent channel access patterns and performed simulations of realistic WLAN application scenarios to determine model parameters. To validate our scheme, we evaluated it via simulations on various such channels. Our future work involves experimenting with different channel

[6] See line 6 of our algorithm.

[7] See line 3 of our algorithm.

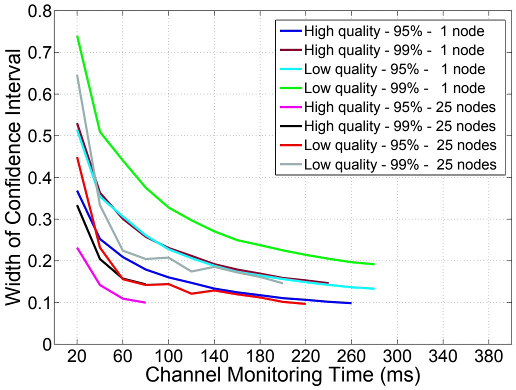

Fig. 4. CI width as a function of the channel monitoring time. The channel corresponds to the case when all wireless nodes are running VoIP.

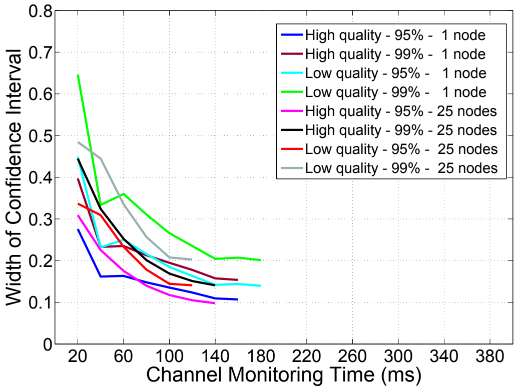

Fig. 5. CI width as a function of the channel monitoring time. The channel corresponds to the case when all wireless nodes are running both FTP and VoIP simultaneously.

TABLE IV
FTP SCENARIO

Monit. **Station** **Quality**	**Conf.** **Lev.** (%)	**Numb.** **of** **nodes**	**Confidence** **Interval** (%)	**Mean** **Load** (%)	**True** **Load** (%)	**Total** **Time**
High	95	1	77.41 - 86.59	82.00	79.15	200
High	99	1	72.76 - 87.24	80.00	78.79	180
Low	95	1	74.12 - 88.10	81.11	78.79	180
Low	99	1	71.06 - 91.16	81.11	78.79	180
High	95	25	75.77 - 85.23	80.50	81.73	200
High	99	25	72.13 - 86.76	79.44	81.76	180
Low	95	25	75.22 - 86.11	80.67	82.64	300
Low	99	25	72.83 - 88.51	80.67	82.64	300

TABLE V
VOIP SCENARIO

Monit. **Station** **Quality**	**Conf.** **Lev.** (%)	**Numb.** **of** **nodes**	**Confidence** **Interval** (%)	**Mean** **Load** (%)	**True** **Load** (%)	**Total** **Time**
High	95	1	28.15 - 38.00	33.08	38.20	260
High	99	1	24.77 - 39.40	32.08	37.60	240
Low	95	1	25.47 - 38.81	32.14	38.36	280
Low	99	1	22.54 - 41.74	32.14	38.36	280
High	95	25	87.51 - 97.49	92.50	88.52	80
High	99	25	85.32 - 99.68	92.50	88.52	80
Low	95	25	85.16 - 94.84	90.00	89.04	220
Low	99	25	82.69 - 97.31	90.00	89.14	200

TABLE VI
MIXED TRAFFIC SCENARIO

Monit. **Station** **Quality**	**Conf.** **Lev.** (%)	**Numb.** **of** **nodes**	**Confidence** **Interval** (%)	**Mean** **Load** (%)	**True** **Load** (%)	**Total** **Time**
High	95	1	74.66 - 85.34	80.00	77.12	160
High	99	1	72.31 - 97.69	80.00	77.12	160
Low	95	1	74.12 - 88.10	81.11	77.10	180
Low	99	1	71.06 - 91.16	81.11	77.10	180
High	95	25	81.54 - 91.32	86.43	86.06	140
High	99	25	79.39 - 93.47	86.43	86.06	140
Low	95	25	81.30 - 95.37	88.33	85.91	120
Low	99	25	78.21 - 98.46	88.33	85.91	120

access patterns and evaluating our mechanism across a wider variety of WLAN settings.

ACKNOWLEDGEMENT

We thank Maria Papadopouli and the anonymous reviewers for their valuable comments.

REFERENCES

[1] S. Mangold, Z. Zhong, G. R. Hiertz, and B. Walke, “IEEE 802.11e/802.11k Wireless LAN: spectrum awareness for distributed resource sharing,” *Wireless Communications and Mobile Computing*, vol. 4, no. 8, pp. 881–902, 2004.

[2] S. Mangold, Z. Zhong, K. Challapali, and C.-T. Chou, “Spectrum agile radio: radio resource measurements for opportunistic spectrum usage,” in *Proc. IEEE Global Telecommunications Conference (GLOBECOM ’04)*, vol. 6, 29 Nov.-3 Dec. 2004, pp. 3467–3471.

[3] M. S. Gast, *802.11 Wireless Networks: The Definitive Guide*, M. Loukides, Ed. Sebastopol, CA, USA: O’Reilly & Associates, Inc., 2002.

[4] S. Mangold and L. Berlemann, “IEEE 802.11k: improving confidence in radio resource measurements,” in *Proc. 16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2005)*, vol. 2, Sep. 2005, pp. 1009–1013.

[5] E. N. Gilbert, “Capacity of a burst-noise channel,” *The Bell System Technical Journal*, vol. 39, pp. 1253–1265, September 1960.

[6] S. M. Ross, *Stochastic Processes*. New York: Wiley, 1983.

[7] A. O. Allen, *Probability, Statistics and Queueing Theory with Computer Science Applications*. Academic Press, 1990.

[8] D. A. Levine, “Student’s *t*-distribution,” *Commun. ACM*, vol. 12, no. 1, pp. 37–38, Jan. 1969.