

Models for Non-intrusive Estimation of Wireless Link Bandwidth

Jian Zhang¹, Liang Cheng², and Ivan Marsic¹

¹ Rutgers University, Department of Electrical and Computer Engineering
Piscataway NJ 08854, USA

{janz, marsic}@caip.rutgers.edu

WWW home page: <http://www.caip.rutgers.edu/disciple/>

² Lehigh University, Department of Computer Science and Engineering
Bethlehem, PA 18015, USA

cheng@cse.lehigh.edu

Abstract. Dynamics of link bandwidth of a wireless link, which changes frequently and abruptly due to the dynamic channel sharing, fading, and mobility, is of interest to adaptive network applications and communication protocols. This paper presents a novel approach to estimate wireless link bandwidth based on radio signal-to-noise ratio (SNR). Unlike traditional methods that send probe packets, our method is non-intrusive to the wireless network since in IEEE 802.11 wireless local area networks, SNR information is provided by the physical layer for the MAC- and upper layers' functionality. Theoretical analysis and experimental observation indicate a nonlinear relationship between SNR and the wireless bandwidth. Based on this, nonlinear models using neural network and Bayesian inference methods are proposed and evaluated on data collected in 802.11b wireless networks. The effectiveness of our method under various environments and scenarios has been studied.

1 Introduction

Adaptation is widely recognized as key to overcoming the resource constraints in mobile computing systems [1, 9]. Only through effective adaptation can mobile applications overcome mobile resource constraints. A key mobile resource is wireless link bandwidth, and its measurement is important in order to devise effective adaptation techniques. Due to dynamic channel sharing, fading, and user mobility, the bandwidth of a wireless link changes frequently and abruptly. Since wireless link generally exhibits characteristics such as large transmission latency, high packet loss rate, and dynamic link bandwidth, it is challenging to estimate wireless link bandwidth accurately and efficiently.

There exist a number of methods for bandwidth estimation. Most of them can be categorized into two groups [12]. One group is a variant of pathchar [5, 8] based on the round-trip times for single packets. The disadvantage of this group is heavy overhead of the probing traffic. The other is a variant of packet-pair [2, 3, 13, 16, 17] based on the packet-pair model. This group imposes lower overhead

relative to the first one. In [11], a multi-packet model is presented to unify the one-packet and packet-pair model and a packet-tailgating technique is proposed.

However, these methods are intrusive to the network since they introduce overhead traffic by sending probe packets during the estimation process. The overhead is not desirable, especially in wireless communication environments where network bandwidth and battery energy are scarce and precious. In addition, due to the dynamics of the wireless link, there is a need for more frequent bandwidth measurement, thus consuming even more resources than in wired networks. Therefore, a non-intrusive bandwidth estimation method is required for wireless networks.

In the existing work on non-intrusive or passive methods, e.g., [11], instead of sending own probe packets, the traffic of other applications running on the same host is observed. However, these methods are inaccurate because they cannot control the size and transmission time of the packets. E.g., [11], estimates the bandwidth based the time interval between packet pair arrivals, which may be inaccurate for small packets due to the coarse granularity of the system clock, thus degrading the estimation performance.

Our objective is to monitor the wireless link as it appears on top of the MAC layer. *Wireless link bandwidth* is defined as the effective transmission bandwidth of a wireless link or *saturated throughput* that can be achieved at the wireless link. The techniques reviewed above measure the nominal or “ideal” channel bandwidth, since they do not account for high error rate and lengthy retransmission time in the MAC layer.

In this paper we propose a non-intrusive method that uses RF signal-to-noise-ratio (SNR) information to dynamically estimate the wireless link bandwidth. The SNR information is provided by most IEEE 802.11 [7] wireless card drivers for the MAC layer and upper layers’ functionality. It is updated by each received frame, which is not necessarily a data frame, but could also be a management or control frame. Another salient characteristic of our method is that the estimation accuracy is not influenced by packet sizes, compared to other traditional methods such as packet-pair techniques.

The radio propagation channel exhibits many different forms of channel impairments [18]. Common ones include distortion due to multipath delay spread, Doppler spread, RF (intracell and intercell) interference, shadow fading, ambient noise, etc. It is to be expected that the wireless bandwidth will follow the variations in the received signal power. Experimental results for an IEEE 802.11b network shown in Fig. 1(a) corroborate this intuition. The SNR is read from the network interface card and the actual bandwidth is measured by sending saturated probe traffic, as explained in Section 3 below. Fig. 1(b) indicates that the relationship is nonlinear. The high bandwidth variability visible in Fig. 1(a) is due to the rapid variations in the channel characteristics, which cause packet loss and retransmission. The high spikes represent the nominal bandwidth, i.e., without errors and MAC-layer retransmissions; the low spikes represent the worst case of errors and retransmissions.

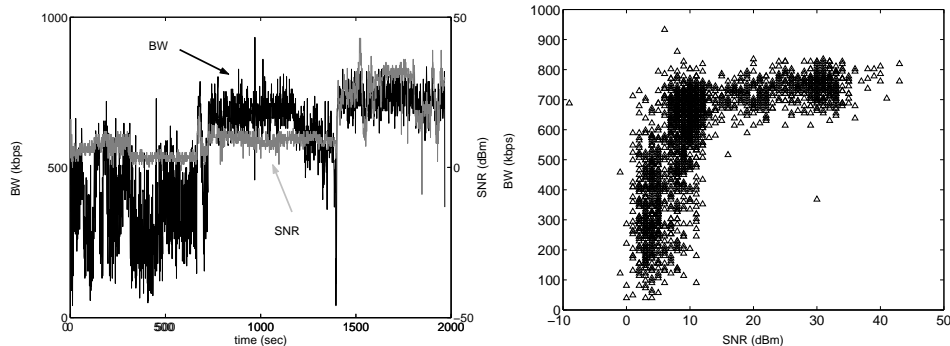


Fig. 1. (a) Example of measured wireless link bandwidth and SNR data. (b) SNR-bandwidth relationship for a typical dataset clearly shows nonlinearity.

Note that our method provides a single-hop rather than end-to-end bandwidth, because SNR only reveals the quality of the immediate wireless link. Nonetheless, this is sufficient for the following reasons. In infrastructure networks, the wireless links are almost always located in the first or last hops. In such networks, the wireless-hop link is usually the weakest link, and its bandwidth information is critical for providing QoS and improving end-to-end performance in such networks [15]. A piecewise mechanism, which considers separately wireless and wired networks, can also be employed [4]. In mobile ad-hoc networks, knowing the bandwidth of individual links is sufficient to determine the best routing path. Link bandwidth information can be exchanged in link-state packets along with topology information.

The rest of the paper is organized as follows. Section 2 presents the theoretical model that shows a relationship of the wireless link bandwidth and SNR. Section 3 describes experiments to collect the data for the empirical bandwidth estimation models and Section 4 presents two empirical models. The effectiveness of the models under different scenarios and environments is discussed in Section 5. Lastly, Section 6 concludes the paper.

2 Theoretical Relationship between SNR and Wireless Link Bandwidth

Here we first demonstrate analytically the relationship between the SNR and wireless link bandwidth. The analysis is based on baseband SNR, which is different from what the 802.11 wireless device drivers provide, i.e., SNR of RF signals. The difference does not invalidate the derived relationship, though.

Signal-to-noise ratio (SNR) in analog and digital communications is a measure of signal strength relative to the background noise. If the signal strength in microvolts is V_s , and the noise level, also in microvolts, is V_n , then the SNR, in decibels, is given as $\text{SNR} = 20 \log_{10}(V_s/V_n)$. A corresponding measure in a

wireless communication environment is the received bit-energy-to-noise ratio, denoted as $\text{SNR}(t) = \alpha(t)^2(E/N)$, where $\alpha(t)$ is a Rayleigh-distributed random process in the flat fading channel model [18], E is the average bit energy of the received signal, and N is the noise power density.

The bit error rate (BER) is the percentage of bits in error relative to the total number of bits received in a transmission. In the wireless communications, the relationship between BER and SNR depends on the channel modulation scheme [10], such that $\text{BER}(t) = Q[\text{SNR}(t)]$, where $Q[\cdot]$ is a nonlinear operator. E.g., in the $(\pi/4)$ QPSK modulation scheme, $Q[\cdot]$ can be expressed as

$$Q(p) = 1\sqrt{2\pi} \int_{\sqrt{2p}}^{\infty} e^{-\frac{x^2}{2}} dx$$

The relationship between packet error rate (PER) and BER depends on the channel coding scheme. Assume that there is no error-correction coding applied and the number of bits in a packet is n , then

$$\text{PER}(t) = 1 - \prod_{i=1}^n [1 - \text{BER}(t_i)]$$

where t_i is the time instant the receiver received the i^{th} bit. If the wireless channel fading variation is very slow compared to the packet transmission time (true for walking and driving speeds), then PER can be approximated as $\text{PER}(t) = 1 - [1 - \text{BER}(t)]^n$. Thus $\text{PER}(t)$ gives the probability of packet loss at time t on the wireless channel and can be viewed as $\text{PER}(t) = P[\text{BER}(t)]$, where $P[\cdot]$ is a nonlinear operator.

The relationship between the wireless link bandwidth and PER depends on MAC layer protocols, such as automatic retransmission (ARQ) scheme. Generally a packet is kept in the transmitter queue until it is successfully received and acknowledged by the receiver. According to the ARQ scheme, a new packet will not be sent to the receiver until either (i) a timer times out, or (ii) the previous packet has been successfully received and acknowledged by the receiver and the sender received the acknowledgement. Therefore the channel efficiency η , which is the percentage of time the channel is used to effectively communicate data between the transmitter and the receiver, can be derived as follows.

Assume that t_p and t_a are the times to transmit a packet and to transmit an ACK, respectively. Furthermore, t_{proc} and t_{prop} are the packet processing time at the end-hosts and the packet propagation time across the channel, respectively. Let t_{to} and t_s denote timeout value of the timer and the minimum time between successive packet transmissions at the sender in case of no packet error, respectively. Both t_{to} and t_s are set differently for different ARQ schemes. Then define t_s^* as the maximum value of t_s across all ARQ schemes, which is $t_s^* = t_p + 2t_{proc} + 2t_{prop} + t_a$. Define x as the total time taken for a successful transmission of a packet and its ACK, which is a random variable depending on the dynamics of the PER. Thus,

$$E[x] = (1 - PER)t_s + PER(t_{to} + E[x]) = t_s + (PER \times t_{to}) / (1 - PER)$$

The channel efficiency can be expressed as follows:

$$\eta(\text{ARQ}, \text{PER}) = \frac{t_p}{E[x]} = \frac{(1 - \text{PER})t_p}{(1 - \text{PER})t_s + \text{PER} \times t_{to}} \quad (1)$$

For stop-and-wait (S&W), go-back- n (GB n), and selective repeat with infinite buffer (SRP-infinite) schemes, the values of the parameters are set according to the Table 1.

Table 1. Parameters for channel efficiency.

ARQ scheme	t_s	t_{to}	η
S&W	t_s^*	t_s^*	$(1 - \text{PER}) \times t_p / t_s^*$
GB n	t_p	$n \times t_p$	$(1 - \text{PER}) / (1 - \text{PER} + n \times \text{PER})$
SRP-infinity	t_p	t_p	$1 - \text{PER}$

The above analysis assumes that the transmitter sends a frame to the receiver once the communication channel is detected as idle. However, in some protocols, such as IEEE 802.11, after the transmitter detects the idle wireless channel, it still has to wait for a random period of time before sending out the frame. Thus (1) can be generalized as:

$$\eta(\text{ARQ}, \text{PER}) = \frac{t_p}{E[x]} = \frac{(1 - \text{PER})t_p}{t_0 + (1 - \text{PER})t_s + \text{PER} \times t_{to}}$$

where t_0 is the random time before the transmitter sends out the frame during the idle channel period. Then the *wireless link bandwidth* can be expressed as:

$$B(t) = \eta[\text{ARQ}, \text{PER}(t)] \times C$$

where C is the wireless link capacity. The relationship between the wireless bandwidth and the signal strength can be denoted as (2) and it can be simplified as (3):

$$B(t) \xleftrightarrow{\eta^{[\cdot]}} \text{PER}(t) \xleftrightarrow{P^{[\cdot]}} \text{BER}(t) \xleftrightarrow{Q^{[\cdot]}} \text{SNR}(t) \quad (2)$$

$$B(t) = f[\text{SNR}(t)] \quad (3)$$

where $f[\cdot] = \eta[P[Q[\text{SNR}(t)]]]$ or $f = \eta \circ P \circ Q$. Therefore, (3) shows that there is a well-defined relationship between the SNR and the wireless link bandwidth.

Since the equation from RF signal SNR to BER depends on how the wireless receiver implements components of synchronization, interference handling, equalization, and demodulation, it is not possible in practice to build the exact theoretical model for bandwidth estimation. For this reason, below we consider empirical methods for bandwidth estimation. Nonetheless, the above analysis is important to back our intuition about the existence of the relationship.

3 Experiments for Training-Data Collection

Before presenting the empirical bandwidth estimation models, we first describe the experiments for collecting the data that will be used to train the models and verify the estimates. Fig. 2 shows the layouts of our experimental IEEE 802.11b wireless/mobile LANs. In the indoors case (Fig. 2(b)), the mobile laptop travels around the fixed wireless sender along the hallway and enters the offices and the labs. During each session a dataset of about 500 samples is recorded. The maximum distance between the receiver and the sender reaches 100 feet. Generally, there is no line-of-sight (LOS) communication between the sender and the mobile laptop. In the outdoors case (Fig. 2(c)), the mobile receiver travels with varying speeds around the sender with a line-of-sight wireless connection.

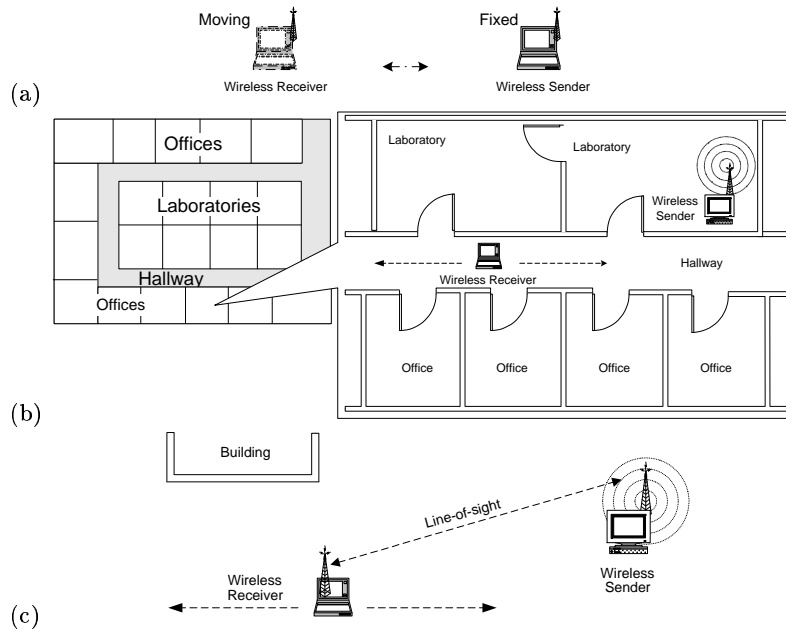


Fig. 2. Layouts of the wireless network for experiments. (a) Topology of the wireless/mobile LAN. (b) Physical layout of the wireless/mobile LAN (Indoors case). (c) Physical layout of the wireless/mobile LAN (Outdoors case).

The experiments were performed in the peer-to-peer (ad-hoc) mode for 802.11 wireless LAN. About 20 datasets of SNR and corresponding wireless link bandwidth recorded at the mobile receivers have been collected. Various IEEE 802.11b wireless cards are used in the experiments, such as Sony VAIO, Orinoco Gold, and Compaq WL 110 cards. The times of day for the data collection are randomly chosen over the period of experiments. To avoid other traffic influencing the recordings, our testbed is isolated and has only one sender and one receiver

operating on a fixed channel. The recordings here are based on one single transmission rate that has been set manually to 1Mbps. In reality, the raw bit rate is automatically set to a level of 1 Mbps, 2 Mbps, 5.5 Mbps or 11Mbps according to current signal quality. The transmission rate is changed by switching among different modulation schemes. We briefly comment upon this in Section 5 below.

Most of the 802.11b wireless cards provide signal strength and noise power independently. The noise power is measured as silence level, i.e. the power measured from the receiver’s antenna at the time when there is no packet coming in. The SNR value we use is calculated by subtracting noise power from the signal power.

During the data collection process, wireless link bandwidth is measured by an intrusive method, which sends probe packets. Since our objective is to monitor the bandwidth of the wireless link, the methods for wired bandwidth measurement reviewed in the introduction would not work here, since they measure the nominal bandwidth. To be as close as possible to the MAC layer, the probe traffic must be user datagram protocol (UDP) packets. We use a constant bit rate (CBR) traffic generator [14] that runs on a fixed wireless sender and keeps sending probe traffic to the wireless receiver. To measure the maximum throughput that reflects the wireless link bandwidth, we saturate the link by setting the CBR traffic rate a slightly higher rate than the physical capacity of the wireless link. Every second, the wireless receiver reads SNR from the wireless card driver. At the same time, it calculates the *saturated throughput* by multiplying the number of packets it has received during the 1 sec period with the sum of the sizes of IP and UDP headers and the UDP payload. This is recorded as the actual link bandwidth for a given SNR. An example measurement is shown in Fig. 1(a). A 1-second sampling period is chosen only for computational convenience. While doing estimation, the sampling rate can be higher than once per second depending on the requirements of the application and it can increase up to the rate of updating the SNR information, i.e., the frame arrival rate. The recorded SNR and bandwidth data are used to perform model identification.

4 Empirical Nonlinear Models

System identification is about modeling the relationship between inputs and outputs. When building a model for recorded datasets, we try to build a model such that the relationship of its output (estimated bandwidth) to its input (SNR) matches what is exhibited by most of the points in the datasets. When the model is used in estimation, real-time measured SNRs are fed into the model and the generated outputs are estimates of bandwidth. In order to evaluate the accuracy of the model, the estimated values are compared to the actual bandwidth, which is again measured by sending probe packets as in Section 3 above.

We use absolute mean error and average relative error as indices for performance evaluation. The average relative error is defined as $(\sum |y_i - \hat{y}_i|) / \sum y_i$, where y_i and \hat{y}_i are the desired and estimated outputs at i^{th} point, respectively. Since both the empirical data (Fig. 1(b)) and the analytical model (Eq. (3)) indi-

cate that the relationship is nonlinear, we employ nonlinear estimation methods. Two methods, neural networks and statistical inference, are described below.

4.1 Back-Propagation Neural Network (BPNN)

Back-propagation neural network (BPNN) has been applied successfully to solve some difficult problems of input-output nonlinear mapping, curve fitting, etc. [6]. For BPNN, the nonlinear function $f(u)$ consists of a series of units combining linear operations (weights and bias) and nonlinear operations (sigmoidal functions). These units are called perceptrons and organized layer by layer. The network is trained in a supervised manner with an algorithm known as the error back-propagation algorithm based on error-correction learning rule. The training process is illustrated in Fig. 3, where $f(u)$ is given as an initial model. A set of input values and corresponding desired output values are used as a training dataset. The difference between desired output values and the output of $f(u)$ is considered an error and propagated back to the model. Based on the errors, the learning algorithm adjusts the parameters in function $f(u)$ and finally minimizes the sum of square errors of all training points, i.e., $\sum (y_i - \hat{y}_i)^2$.

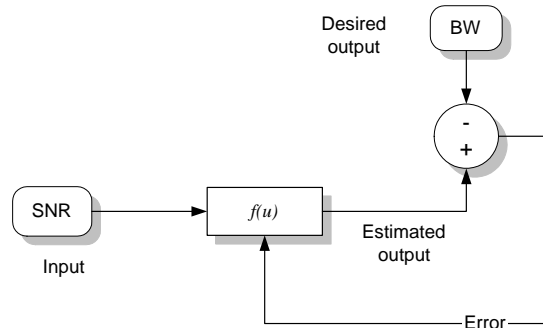


Fig. 3. Error-correction learning model for neural networks.

The reason we choose BPNN for nonlinear modeling is that the relationship of measured bandwidth vs. SNR, shown in Fig. 4, resembles a sigmoidal function. Therefore, with a simple structure, i.e., small number of perceptrons and layers, and a short training time, the BPNN can learn to represent the relationship. Moreover, the computation complexity of the estimation is also low. We may notice in Fig. 4 that the SNR-to-bandwidth mapping is not one-to-one, i.e., there could be different bandwidth values corresponding to a single SNR value. Given a set of training points, a BPNN learns a one-to-one sigmoidal mapping that minimizes the sum of square errors.

In this paper, a 1-4-1 BPNN is used for modeling, i.e., one input, 4 hidden nodes, and 1 output. The model is first trained on a certain dataset and then it is

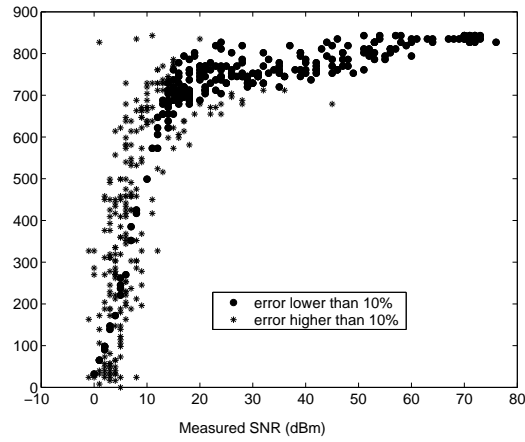


Fig. 4. Least sum of square errors fitting obtained by BPNN.

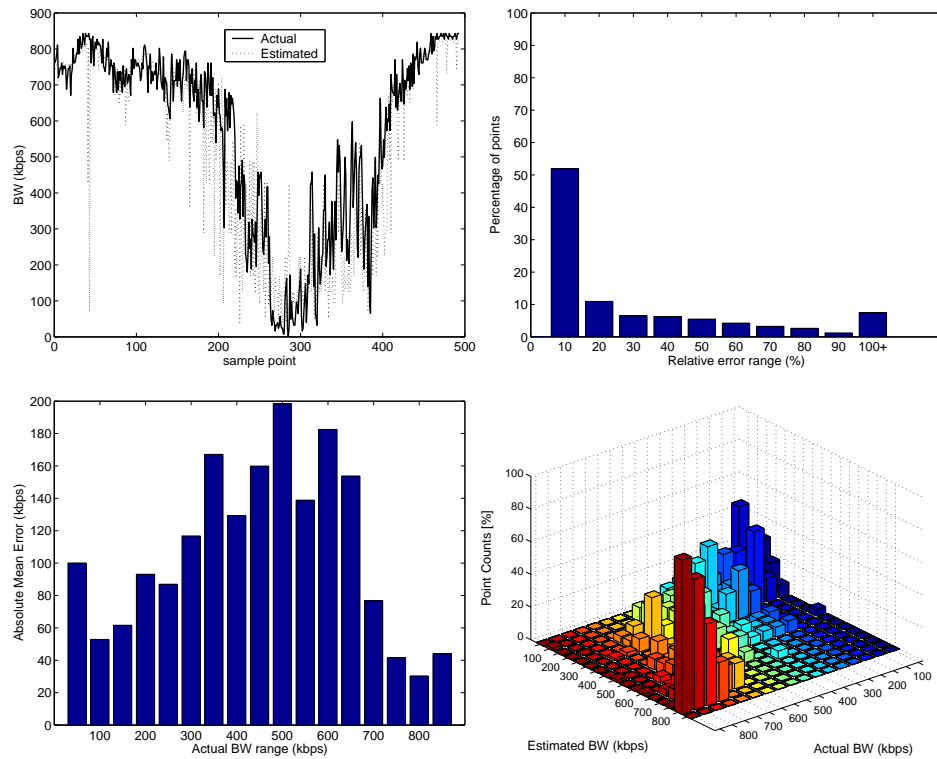


Fig. 5. Performance of BPNN model: (a) Comparison of measured and estimated bandwidths; (b) The distribution of relative errors; (c) Mean absolute error for estimations; (d) 2-D histogram.

used to generate estimates by feeding other datasets of recorded SNRs. The estimates on each dataset are compared with the corresponding actual measured link bandwidth, and errors are presented in the form of relative and absolute mean value. This is shown in Fig. 5. The relative error between the estimated bandwidth and the actual measured bandwidth in Fig. 5(a) is 14.81%. Fig. 5(b) shows the distribution of estimates in different relative error ranges. We may notice that for about 50% of estimates relative error is less than 10%. For very low bandwidths, even a small absolute error means a large relative error. Therefore, Fig. 5(c) shows the mean absolute errors for different values of actual bandwidth. The largest mean absolute error is around 200 Kbps and it is located in the bandwidth range from 320 Kbps to 600 Kbps, which means bandwidths in that region are more difficult to track by SNRs. Lastly, Fig. 5(d), shows the model performance as a 2-D histogram. Each bar counts the number of different estimates for a given actual bandwidth. The bars close the diagonal (estimated = actual) are the tallest, indicating that the model works well for this dataset.

We repeated testing the BPNN model by training it on 5 datasets and applied it to do estimation on 5 different datasets. The average relative error obtained is 24.94% and the standard deviation of error is 13.57%.

4.2 Statistical Model: Bayesian Inference

Considering the collected SNR data as observations and the corresponding measured link bandwidth as output states, we can build a Bayesian inference model. Firstly, we divide the ranges of input data and output data into small regions, and represent them by a finite state space s_1, s_2, \dots, s_n and an observation space o_1, o_2, \dots, o_m . Then, based on the historical data, a Bayesian inference model is built by calculating the priority distribution: $Pr(o_i|s_j)_{i=1,2,\dots,m;j=1,2,\dots,n}$. This gives the probability of a certain SNR value at region o_i observed in the condition that real or measured bandwidth is at state s_j , based on collected data.

To perform the estimation, we use the following Bayes rule to update the probabilities, which represent the new level of belief in all possible outcome states after an observation:

$$Pr(s_j|o^*)_{j=1,2,\dots,n} = \frac{Pr(s_j)Pr(o^*|s_j)}{\sum_{i=1}^n Pr(s_i)Pr(o^*|s_i)}$$

where o^* is the current observation; $Pr(s_j)$ is the prior probability and it is initialized by assuming the each state has the same probability in the beginning.

The output of the Bayesian inference model provides information of the probability distribution of all possible states of estimated bandwidth. We calculate the expected value of the probability distribution and take it as the estimated bandwidth. Similar to the BPNN model, one dataset with both measured SNR and measured bandwidth is used to train the model, i.e., to build the priority distribution. Then, the estimations are made on other datasets and result compared with the measured actual bandwidth. The training and estimation datasets used to obtain Fig. 6 are the same as the ones used with the neural network model. Here, the relative error is 15.70%.

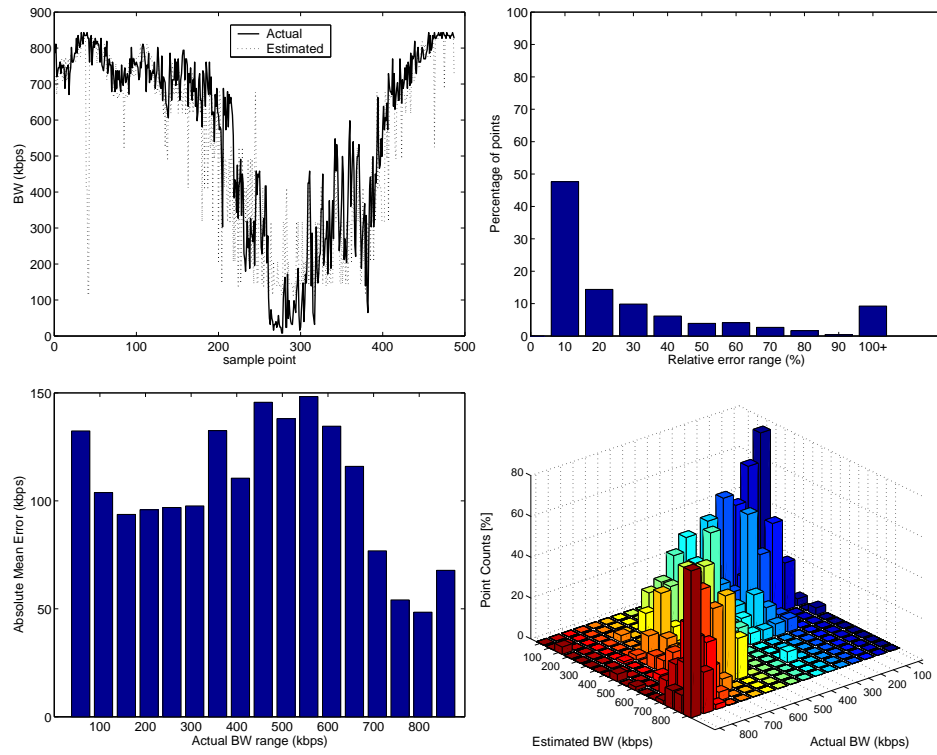


Fig. 6. Performance of the Bayesian inference model: (a) Comparison of measured and estimated bandwidths; (b) The distribution of relative errors; (c) Mean absolute error for estimations; (d) 2-D histogram.

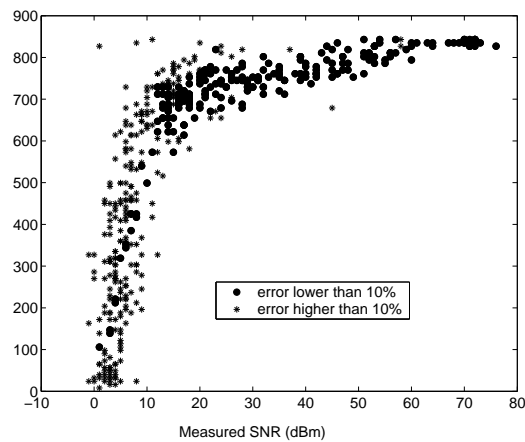


Fig. 7. The Bayesian model mapping resembles the least sum of square errors fitting.

We repeated testing the Bayesian model by training it with 5 datasets and applying it to do estimation on 5 other datasets. The average relative error obtained is 26.07% and standard deviation of error is 15.43%. This is close to what we obtained with the BPNN model. Fig. 7 explains the similarity, since the mapping curve obtained by the Bayesian model resembles the least-square-error curve obtained by the BPNN in Fig. 4.

The performances of these two models are compared in Table 2. In the columns list the training datasets and the rows list the estimation datasets. Relative errors caused by BPNN model and Bayesian model corresponding to different datasets are shown respectively in the upper line and lower line of each cell. The table shows their performances parallel each other.

Table 2. Performance comparison of neural network and Bayesian inference models. Shown are the error rates for each. The performance on some of the datasets is significantly worse than on others, e.g., Test data III. This is since the datasets are not balanced, so some datasets contain more points in the “bad” region (SNR < 15 dBm).

BPNN Bayes	Test data I	Test data II	Test data III	Test data IV	Test data V
Training data I	15.32% 16.08%	15.98% 17.80%	44.69% 56.22%	22.74% 24.06%	22.46% 21.03%
Training data II	15.91% 15.96%	15.48% 16.18%	45.31% 49.97%	24.09% 21.83%	23.66% 20.62%
Training data III	15.48% 15.70%	15.82% 16.18%	46.85% 49.97%	23.62% 21.83%	23.77% 20.62%
Training data IV	14.69% 15.87%	16.00% 19.77%	52.28% 60.05%	19.39% 18.78%	20.54% 18.98%
Training data V	15.15% 15.78%	16.10% 20.91%	61.71% 62.02%	17.20% 17.84%	19.20% 17.82%

The performance of both models is particularly weak at the steep part of the SNR-bandwidth curve, for SNRs lower than 15 dBm, Figs. 4 and 7, where the mapping is obviously not one-to-one. We tried to improve this performance by designing the models with multiple input variables. One possibility is to use the signal and noise strengths as two independent inputs, rather than a single SNR input. Another possibility is to use past values of SNR or bandwidth some of their statistical properties as additional input variables for the model. So far these efforts only resulted in minor performance improvements. However, we still believe that the model performance at this part of the curve can be improved only through additional input data and this is a topic of our current research.

It may be argued that these models are suitable only to the extent that they estimate a “break point” between a good and bad link. However, although accurate bandwidth estimation for low SNRs is hard to achieve, having at least average bandwidth estimate is useful for many applications. In addition, for the

case of automatic rate selection (11 Mbps, 5.5 Mbps, 2 Mbps, and 1 Mbps), the “break point” is not as easy to identify since the performance deteriorates rather gradually. Even if it could be located, it would not provide sufficient information for link characterization for bandwidth-sensitive applications.

5 Model Effectiveness Under Different Scenarios

Since both BPNN and Bayesian models are built or trained offline on previously collected data and there is no actual bandwidth information available at runtime when doing estimation, it is important to know whether the environmental changes could cause a performance deterioration. The first experiment was in training the models on data measured in one building and using them for estimation in a different building. The average relative error is 17.6%.

Fig. 8 shows the input-output curves for the models trained on all indoor collected data and all outdoor collected data. Obviously, there is a slight difference and this difference contributes to the performance degradation when using a model trained on one environment for estimation in another one:

<u>Estimation</u> → Training ↓	Indoor	Outdoor
Indoor	24.94%	33.04%
Outdoor	32.98%	31.97%

A possible solution is to employ a dual-model mechanism. When mobile nodes are located outdoors, the system employs the model trained on outdoors data and vice versa for indoors. In this case, it is important to automatically recognize the environment change and switch to a proper model and this is a possible future research topic.

Experiments similar to those reported above for peer-to-peer (ad-hoc) mode were also performed for the infrastructure (access point) mode. The results remain virtually the same if one of the endpoints is connected to the access point over a wire. If both endpoints communicate wirelessly over the access point, the performance changes due to channel sharing and different spatial relationships to the access point. This case needs to be investigated further.

In addition to the original 802.11-DS standard, 802.11HR offers additional 5.5 Mbps and 11 Mbps for high rate transmission [7], where a different modulation (CCK) is adopted. To extend our models to a multi-rate situation, we repeated the experiments with non-fixed transmission rates. Experiment results show that the method performs well in multi-rate case and the average relative error in our tested cases is 18.58%. Future research includes extending the method to other wireless LAN standards, such as 802.11g.

Since our method estimates the link bandwidth at the MAC layer, it can be used with different network or transport protocols, including TCP. However, in such a case, the estimated bandwidth does not necessarily predict the throughput that will be achieved by the transport protocol. We evaluated the estimation

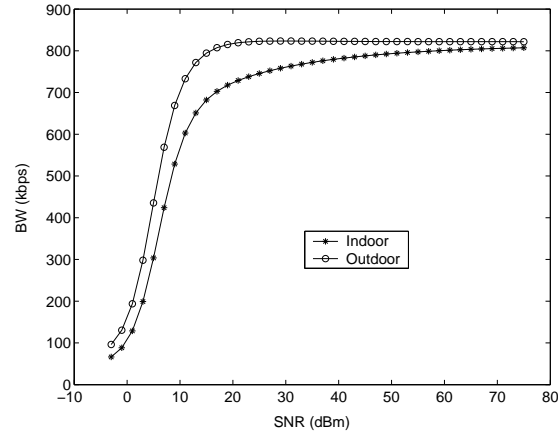


Fig. 8. The least sum of square errors fittings for indoors and outdoors datasets.

models developed here with TCP traffic. The measured TCP throughput sharply falls off for low estimated bandwidth because of the congestion control mechanisms reacting to highly increased packet loss under a low link bandwidth. This relationship needs to be further investigated.

6 Conclusions

The dynamics of the wireless link bandwidth is of interest to adaptive applications and protocols in wireless communication environments. This paper presents a novel approach to estimate wireless link bandwidth based on radio signal-to-noise ratio. Its salient feature is that it provides a passive, non-intrusive method for wireless link bandwidth estimation, which avoids the use of probe packets as in traditional methods. The method is evaluated in IEEE 802.11b wireless local area networks under different environmental conditions. Two nonlinear models, BP neural network and a statistical model, are used and their performance is studied and compared. Both BPNN and Bayesian inference models achieve similar accuracy in bandwidth estimation. Future work includes improving the accuracy of the estimation method under different scenarios and employing it in real-world applications to provide quality-of-service. Another interesting parameter related to signal strength is the link latency, which we also plan to estimate. Also, the scenarios with multiple senders and receivers will be studied.

Acknowledgements

The research is supported by NSF Contract No. ANI-01-23910, US Army CE-COM Contract No. DAAB07-02-C-P301, a research grant from Cisco, Inc., and by the Rutgers Center for Advanced Information Processing (CAIP).

References

1. Badrinath, B., Fox, A., Kleinrock, L., Popek, G., Reiher, P., Satyanarayanan, M.: A conceptual framework for network and client adaptation. *Mobile Networks and Applications (MONET)* **5** (2000) 221-231
2. Bolot, J. C.: End-to-end packet delay and loss behavior in the internet. *Proc. ACM SIGCOMM'93 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications* (San Francisco CA, September 1993) 289-298
3. Carter, R. L., Crovella, M. E.: Measuring bottleneck link speed in packet-switched networks. *Performance Evaluation* **27/28** (1996) 297-318
Also appears as Technical Report TR-96-006, Department of Computer Science, Boston University, Boston, MA, March 1996
4. Cheng, L., Marsic, I.: Piecewise network awareness service for wireless/mobile pervasive computing. *Mobile Networks and Applications (MONET)* **7** (2002) 269-278
5. Downey, A. B.: Using pathchar to estimate internet link characteristics. *Proceedings of the ACM SIGCOMM'99 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Cambridge MA, August/September 1999) 241-250
6. Hertz, J., Krogh, A., Palmer, R. G.: *Introduction to the Theory of Neural Computation*. Addison-Wesley Publ. Co., Reading, MA, 1991
7. IEEE Computer Society, LAN MAN Standards Committee: IEEE Standard 802.11-1999, 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications (1999)
8. Jacobson, V.: Pathchar—A tool to infer characteristics of internet paths. Mathematical Sciences Research Institute (MSRI) Math Awareness Week (Mathematics and the Internet), April 21, 1997
Online presentation and software at: <ftp://ftp.ee.lbl.gov/pathchar>
9. Katz, R. H.: Adaptation and mobility in wireless information systems. *IEEE Personal Communications* **1** (1994) 6-17
10. Kim, Y. Y., Li, S.: Modeling multipath fading channel dynamics for packet data performance analysis. *Wireless Networks* **6** (2001) 481-492
11. Lai, K., Baker, M.: Nettimer: A tool for measuring bottleneck link bandwidth. *Proc. 3rd USENIX Symposium on Internet Technologies and Systems* (San Francisco CA, March 2001) 122-133
12. Lai, K., Baker, M.: Measuring link bandwidths using a deterministic model of packet delay. *Proc. ACM SIGCOMM 2000 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications* (Stockholm Sweden, August 2000) 283-294
13. Lai, K., Baker, M.: Measuring bandwidth. *Proc. Conf. on Computer Communications (IEEE INFOCOM'99)* (New York NY, March 1999) 235-245
14. Naval Research Laboratory (NRL): The Multi-Generator Toolset MGEN-3.2 User's Guide, Washington, DC
Online at: <http://manimac.itd.nrl.navy.mil/MGEN/MgenUserGuide.html>
15. Qiu, L., Bahl, P., Adya, A.: The effect of first-hop wireless bandwidth allocation on end-to-end network performance. *Proc. NOSSDAV'02* (Miami FL, May 2002)
16. Paxson, V.: End-to-end internet packet dynamics. *IEEE/ACM Trans. Networking* **7** (1999) 277-292
17. Paxson, V.: Measurements and Analysis of End-to-End Internet Dynamics. Ph.D. Dissertation, Computer Science, University of California, Berkeley, CA, April 1997
18. Rappaport, T. S.: *Wireless Communications Principles and Practice*. 2nd edition. Prentice-Hall, Englewood Cliffs, NJ, 2002