An Experimental Study of Packet Loss and Forward Error Correction in Video Multicast over IEEE 802.11b Network

Özgü Alay, Thanasis Korakis, Yao Wang, and Shivendra Panwar
Department of Electrical and Computer Engineering, Polytechnic University, Brooklyn, NY
E-mails: ozgu@vision.poly.edu, {korakis,yao}@poly.edu, panwar@catt.poly.edu

Abstract— **Video multicast over Wireless Local Area Networks (WLANs) faces many challenges due to varying channel conditions and limited bandwidth. A promising solution to this problem is the use of packet level Forward Error Correction (FEC) mechanisms. However, the adjustment of the FEC rate is not a trivial issue due to the dynamic wireless environment. This decision becomes more complicated if we consider the multi-rate capability of the existing wireless LAN technology that adjusts the transmission rates based on the channel conditions and the coverage range. In order to explore the above issues we conducted an experimental study of the packet loss behavior of the IEEE 802.11b protocol. In our experiments we considered different transmission rates under the broadcast mode in indoor and outdoor environments. We further explored the effectiveness of packet level FEC for video multicast over wireless networks with multi-rate capability. In order to evaluate the system quantitatively, we implemented a prototype using open source drivers and socket programming. Based on the experimental results, we provide guidelines on how to efficiently use FEC for wireless video multicast in order to improve the overall system performance. We show that the Packet Error Rate (PER) increases exponentially with distance and using a higher transmission rate together with stronger FEC is more efficient than using a lower transmission rate with weaker FEC for video multicast.**

Index Terms: forward error correction, packet error rate, wireless networks, IEEE 802.11b, video multicast

I. INTRODUCTION

In recent years, the demand for video applications over wireless networks has risen with the increase in both the bandwidth of wireless channels and the computational power of mobile devices. Multicasting is an effective solution for simultaneous transmission of data to a group of users, since it saves network resources by spreading the same data stream across multiple receivers. However, the high error rate of the wireless channel, along with heterogeneity of the users, make multicast of real time services over wireless networks a challenging problem.

In a wireless network, as the external environment changes, the channel error rate varies, resulting in devastating effect on multimedia transmission. In order to cope with errors and hence have robust video transmission, we need accurate channel-condition estimation and an effective error control mechanism. Furthermore, due to bursty and location dependent errors, each user in a multicast system will most likely lose different packets. Therefore, a simple ARQ (Automatic Repeat reQuest) based scheme is not appropriate for video multicast over wireless channels since it can cause a large number of retransmissions.

A promising solution for error control in video multicast over wireless networks is the use of forward error correction (FEC) to handle losses. In such a system, block erasure codes are used to correct errors using redundant information in the data stream. For example in an (n, k) block erasure code, there are a total of n packets where k of them are source packets and $(n-k)$ of them are redundant parity packets. The parity packets are generated in such a way that any k of the n encoded packets are sufficient to reconstruct the k source packets. The advantage of using block erasure codes for multicasting is that a single parity packet can be used to correct independent single-packet losses among different receivers.

Several studies have shown the efficiency of FEC via simulations and have proven that such a scheme is promising for error correction in wireless multicasting [1]-[4]. Villalon et al. [5] studied a cross-layer approach for adaptive video multicast considering the multi-rate capabilities of wireless networks. Although these simulation results provide some insights on the way FEC should be applied, they do not consider a real wireless network with multi-rate capabilities. Limited implementation approaches in the literature focus on specific algorithms, and therefore they do not present a thorough investigation of the various trade-offs. McKinley et al. [6] studied proxy-based adaptive FEC for reliable multicast in WLANs. They proposed an adaptive FEC mechanism where the number of parity packets transmitted is based on the current data loss rate with a feedback system. The same group extended their studies in [7] and show that combining forward and backward error control is an effective strategy for proxy-based video multicast. In both papers they evaluate the proposed schemes by implementing them in a real testbed. However, their studies considered only an indoor environment and fixed transmission rates (2 Mbps in [6] and 11 Mbps in [7]).

In this paper we study the dynamics of FEC mechanisms for a real wireless 802.11b network. We set up an experimental testbed and studied the behavior of the network in terms of packet error rates for different environment (indoors and outdoors), different distances between the source and destination and different transmission rates. Taking it one step further, based on the results we collected during the above experiments we implemented a FEC system using open source drivers and socket programming. We ran extensive experiments in order to understand the dynamics of FEC when it is applied in a real environment. Based on the results of our experiments we

---

[1] This work is supported in part by National Science Foundation (NSF) under award 0430885, and the New York State Center for Advanced Technology in Telecommunications (CATT). The work is also supported by Wireless Internet Center for Advanced Technology (WICAT), an NSF Industry/University Research Center at Polytechnic University.

provide insights and guidelines on how to practically use FEC schemes for robust wireless video multicast.

The paper is organized as follows. In Section II, packet level FEC is discussed along with the rate adaptation. The implementation effort is elaborated in Section III. We specify the configuration of the experiments in Section IV. Section V reports and analyzes the obtained results. We conclude the paper in Section VI.

II. RATE ADAPTATION WITH PACKET LEVEL FEC

Although rate adaptation is a standard feature in today's wireless networks, multicast/broadcast packets are always transmitted using the base transmission rate of the system (e.g., 1Mbps for IEEE 802.11b). The intention of such a conservative approach is to minimize losses at the stations that are located far away from the transmitter, so that they are able to successfully receive and decode the packet.

Forward error correction (FEC) at the application layer is a promising alternative for handling losses in multicast services. The basic idea of FEC is that redundant information is sent a-priori by the source station, in order to be used by the receivers to correct errors/losses without contacting the source. Since CRC-based error detection at the link layer results in the removal of the corrupted packets, many FEC-based protocols try to recover these packets [8]. However, such a scheme introduces overhead since extra parity packets are now transmitted by the source station. The overhead introduced is the number of parity packets to be sent for k source packets. The number of parity packets, m, can be determined as follows:

$$m = kP_E/(1 - P_E) \tag{1}$$

where P_E is the Packet Error Rate (PER). Note that, the level of the overhead depends on the packet error rates, P_E, in the network. Thus, the higher is the packet loss rate, the more parity packets must be transmitted by the server, thus increasing the overhead and reducing the FEC rate, r_{FEC}, which is the ratio of source packets to the total number of packets, $\frac{k}{k+m} = 1 - P_E$.

From the above discussion, we conclude that it is important to have an accurate estimate on the packet error rate, so that just enough FEC parity packets can be applied. In wireless networks, we know that different transmission rates, R_{PHY}, give different PER. Furthermore, due to path loss, for a fixed transmission rate, we observe different PER values at different distances, d. Hence, for a fixed distance and physical transmission rate, the FEC rate, r_{FEC}, can be formulated as follows

$$r_{FEC}(d, R_{PHY}) = 1 - P_E(d, R_{PHY}) \tag{2}$$

Note that the FEC overhead is not the only overhead in the system. In order to cover the other overheads (e.g., headers, etc.), we define the effective data ratio, β, as the ratio of the time spent to transmit the actual payload data to the total transmission time. Note that MAC and IP headers are sent at the selected transmission rate, whereas the physical layer header is always sent at the base rate. Hence, β depends on physical transmission rate. Typical β values will be presented in Section IV.

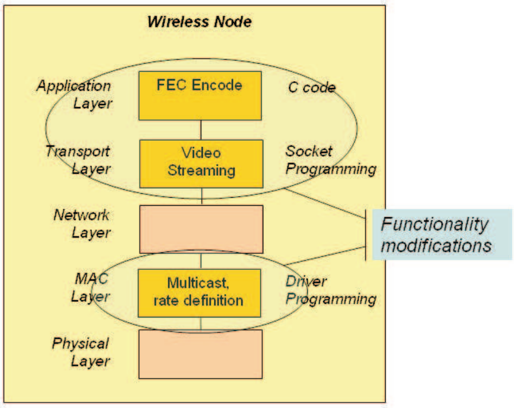


Fig. 1. Node Architecture

Based on the discussion above, the useful rate, R_{useful}, can be computed as follows,

$$R_{useful} = r_{FEC}(d, R_{PHY})\beta(R_{PHY})R_{PHY} \tag{3}$$

In the above formulation, it is not clear how someone should define the combination of transmission rate and FEC rate in order to increase the efficiency of the network. On one hand, the higher the transmission rate is, the higher the PER and therefore the more FEC parity packets should be transmitted. On the other hand, as the transmission rate increases, the more efficient the use of the medium becomes, allowing more room for extra FEC parity packets. Therefore, while designing a multicast system, we should consider transmission rate and the FEC rate jointly.

III. IMPLEMENTATION EFFORTS

Inspired by the above questions we decided to build a real system in order to study the effect of the transmission rate on the packet error rate of a wireless network and define the FEC that should be used. By understanding the interaction between transmission rate, PER, FEC and distance, we can define the guidelines on the way we should combine rate adaptation with FEC in order to improve the efficiency of multicast video services over wireless networks.

In order to implement the system in a way that would allow us to change different parameters and to observe the behavior of different metrics, we changed the basic functionality of a wireless node in three different layers of the protocol stack: the MAC, the transport and the application layer. The block diagram of the system we designed and implemented is depicted in Figure 1. The main features of the system are as follows:

A. MAC layer

For the implementation of the MAC layer we used open source drivers in a Linux platform. In particular we used the MadWifi driver [10] for the Atheros chipsets [11]. We used the driver in broadcast mode (i.e. no acknowledgment, no retransmissions). Additionally, we added a new feature in the driver that allowed us to choose the transmission rate that we would use. In our experiments, we set up the wireless cards to work in the 802.11b mode, and therefore we had to choose between four different rates: 1Mbps, 2Mbps, 5.5Mbps, 11Mbps. For the control of the transmission rate, we built a simple GUI that directly communicates with the driver and allows the user to set up this parameter easily.

B. Transport layer

In order to implement the video streaming service we built a video client/server application using UDP/IP socket programming. The server is a program that can read a FEC encoded video file, packetize accordingly and transmit. Similarly, the program in the client side receives packets, does the FEC decoding and stores the resulting video into a file.

C. Application layer

In the application layer we implement a packet level FEC mechanism. We utilize Reed Solomon (RS) codes since it is one of the well known block codes with good error correction properties and is widely used in FEC schemes. In general an (n, k) RS code contains k source packets and $(n - k)$ parity packets. Altogether, they form a group of n packets, such that any k of the n packets can be used to reconstruct the k source packets [9]. In this work, we use systematic (n, k) codes where the first k of the n encoded packets are identical to the k source packets.

In the current implementation, the generation of the parity packets is done offline. In our experiments we utilize RS(128,64) ($n = 128$, $k = 64$) codes where we generate 64 parity packets for 64 source packets. We store these files in the hard drive of the node in order to use them as inputs on the video streaming server. Note that, although we generate $(n - k)$ parity packets, it would be a waste of bandwidth to send all parity packets. Therefore, the number of parity packets to be sent, m, is chosen based on the channel conditions (i.e. packet error rate P_E) as formulated in Equation (1).

Upon reception of the packets, the receiver decodes the FEC encoded packets, generates the video file and stores it in the node. As long as the number of lost packets is less than m, all original video packets can be decoded successfully. When the loss exceeds the FEC correction limit, only the received video packets are put into the decoded video files.

IV. EXPERIMENTAL SETUP

A. Testbed Configuration

The testbed used in the experiments consists of 2 Linux laptops with 802.11 wireless cards based on the Atheros chipset. The stations share channel 11 of the 2.4 GHz band. In this experimental study, one station is used as a dedicated destination, which mimics the functionality of a receiver and the other station is an access point.

B. Methodology

The experiments we conducted were composed of two parts. For a fixed transmission rate, we first measured the PER using Iperf [12], which is a powerful tool for traffic generation and measurement. In our experimental setup, one of the stations runs an Iperf client to generate UDP traffic streams, while the other runs an Iperf server which receives the traffic and collects the statistics (e.g. PER). To remove any random effect and short-term fluctuation, we ran each experiment 10 times with each run lasting 1 minute. We then averaged the results.

$Packet$ $Size(B)$	N	T_{DATA} $(msec)$	T_{MAC+IP} $(msec)$	T_{PHY} $(msec)$	$Data$ (Mb)	β
1470	161	172	7.25	20.59	1.892	0.172
1000	223	162	10.04	28.51	1.782	0.162
500	374	136	16.86	47.87	1.496	0.136
200	626	91	28.21	80.08	1.001	0.091

TABLE I

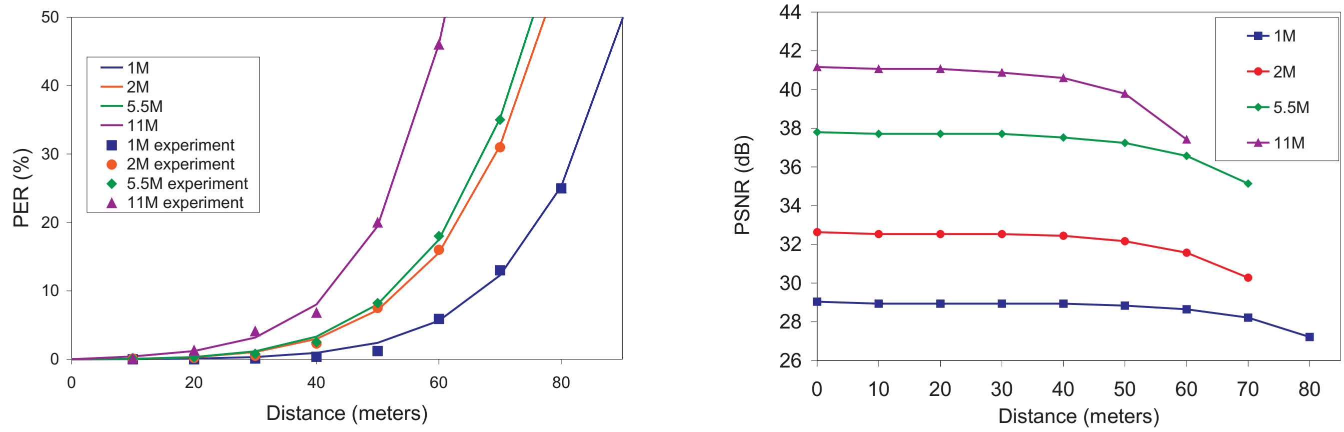
DISTRIBUTION AMONG DATA AND HEADERS FOR 11MBPS FOR DIFFERENT PACKET SIZES(MAC HEADER=34B, IP HEADER=28B (INCLUDING UDP HEADER), PHYSICAL LAYER HEADER=16B)

During the PER measurements, we are mainly interested in the packet losses due to channel conditions rather than the traffic contention in the channel. Hence, in our experiments, we transmit for only 20% of the time, in order to keep the traffic level low. In order to be fair for all transmission rates, we also consider the overhead introduced by MAC, IP and physical layer headers. Note that, MAC and IP headers are sent at the selected transmission rate, whereas the physical layer header is always sent at the base rate, 1Mbps. We also considered the effect of packet size, since the overhead due to headers also depends on the packet size. In Table I, we tabulate the time distribution among data and headers for different packet sizes for 11Mbps physical rate. In this table, for a transmission duration of 1 sec, the total airtime is 200msec. T_{DATA}, T_{MAC+IP}, T_{PHY} denote the time spent for the data, MAC+IP header (including the UDP header) and physical layer header, respectively. N denotes the number of packets that can be transmitted in 200msec and $Data$ illustrates how many bits you can transmit in T_{DATA} msecs at a rate of 11Mbps. Finally, the last column shows the effective data ratio, β. In Table I, we observe that as you increase the packet size, you reduce the overhead due to headers, hence you have more room for the actual data. On the other hand, we also know that as you increase the packet size, you increase the PER. We performed preliminary experiments to investigate the effect of the packet size on the useful rate. We observed that at higher packet sizes, even though we have a higher packet error rate, the useful rate is also higher. Hence, throughout this paper, all the figures are obtained for a packet size of 1470 Bytes.

After the computation of the average packet error rate, we calculated the amount of redundancy needed to correct the errors using Equation 1. We then transmited the parity packets together with the source packets. Here, we arrange the system such that the source packets are transmitted at useful rate. At the receiver side, the received packets are decoded and the generated video is stored. Finally the quality of the video is obtained using the Peak Signal to Noise Ratio (PSNR) of the received video.

V. RESULTS

In our experimental study we use an IEEE 802.11b based WLAN. In order to understand the behavior of such a network we conducted experiments using broadcast modes in an outdoor environment. As described in Section IV, we first obtain PER curves for different physical transmission rates and various locations.



(a) PER vs coverage area

(b) PSNR vs coverage area

Fig. 2. Outdoor performance achievable at different PHY transmission rates (packet size is 1470B)

The outdoor experiments were conducted in Columbus Park, Brooklyn. We ran several experiments for different distances between the access point and the receiver. We varied the distance from 10 to 80 meters. The access point and the receiver are within line of sight. Figure 2(a) illustrates the packet error rate versus distance curve for the broadcast mode. In this figure, the data points are the average loss rate derived from the experimental results, whereas the curves show the exponentials fitted to these results. Here we only illustrate the results up to 50% PER, since for PER higher than 50% we very often lost the connection due to bad channel conditions making the obtained PER values unreliable. The figure shows the basic packet loss behavior of the wireless network. Based on the figure, we observe that,

- As the distance between the access point and the receiver increases, the packet error rate increases exponentially.
- For a target PER, the coverage area (defined as the distance at which the PER is less than the target) reduces significantly as the physical transmission rate increases.

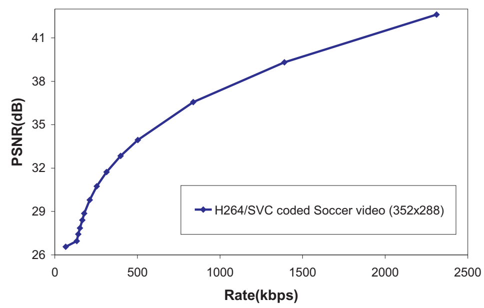


Fig. 3. Rate-distortion curve for Soccer video, obtained by using H.264/SVC encoder [13] using the MGS quality scalability mode with a base layer rate of 65 kbps.

	10m	20m	30m	40m	50m	60m	70m	80m
1Mbps	1	1	1	1	2	4	9	22
2Mbps	1	1	1	2	5	12	30	-
5.5Mbps	1	1	1	3	6	14	35	-
11Mbps	1	1	3	6	16	55	-	-

TABLE II

NUMBER OF PARITY PACKETS NEEDED FOR $k = 64$ SOURCE PACKETS IN BROADCAST MODE(PACKET SIZE IS 1470B)

	10m	20m	30m	40m	50m	60m	70m	80m
1Mbps	187	187	187	187	183	177	163	135
2Mbps	369	369	369	362	344	307	240	-
5.5Mbps	983	983	983	948	890	793	604	-
11Mbps	1857	1857	1791	1700	1455	931	-	-

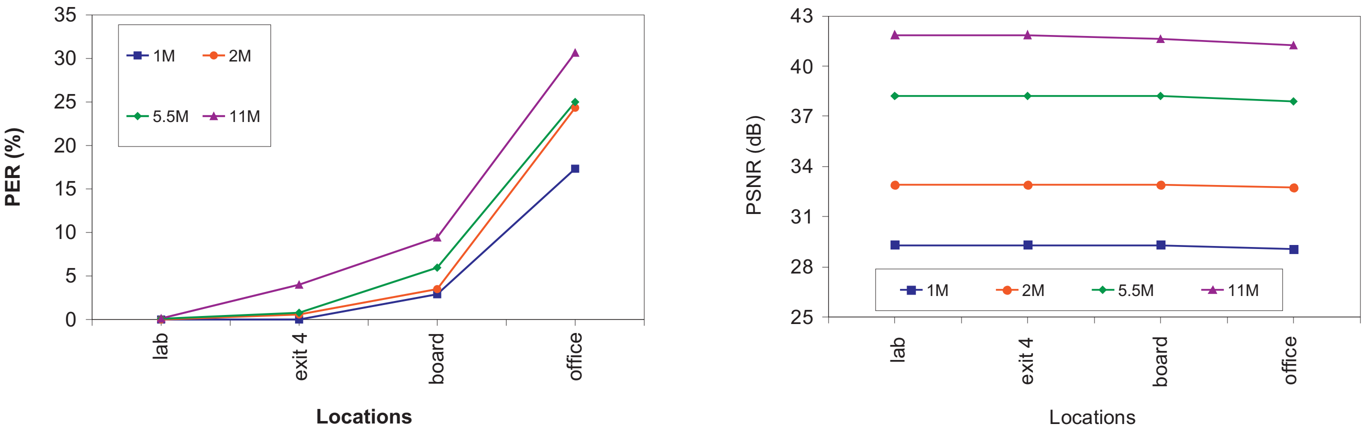
TABLE III

USEFUL RATE IN KBPS FOR BROADCAST MODE (PACKET SIZE IS 1470B)

Based on the PER's obtained, we computed the appropriate number of parity packets. Here note that the PER resulting from the experiments is the average packet error rate among several runs. Thus, in order to cope with the instant channel changes, we transmit an additional 20% parity packets as compared to that computed using Equation (1). From our experiments, we found that using 20% more parity packets than that dictated by the average loss rate can correct practically all lost packets. In Table II, we specify the required number of parity packets for different coverage areas. As seen in the table, for a target distance as we increase the transmission rate, due to higher PER we need to send more parity packets resulting in more overhead. On the other hand, since we are sending at a higher rate, the useful rate (video rate) is increased as tabulated in Table III.

The packet video streams in our experiments are created by encoding a video clip (Soccer, 352x288, 30 Hz) using a H.264/SVC encoder [13]. The PSNR vs. video bit rate curve is shown in Figure 3. By combining Table III and Figure 3, we generate the PSNR vs. distance curves in Figure 2(b) for the broadcast mode in the outdoors. The results show that,

- For distances that are reachable with higher transmission rates, it is better to send at a higher rate with more error protection than to send at a lower rate with less error protection. For example, in Figure 2(b), if the target



(a) PER vs locations (b) PSNR vs locations

Fig. 4. Indoor performance achievable at different PHY transmission rates (packet size is 1470B)

coverage area has a maximum distance of 50 m, and the data is transmitted with the base rate which is 1Mbps you can only achieve a video quality of 28.83dB (2 parity packets resulting in a video rate of 183kbps). However if you chose to transmit at 11Mbps you can achieve a video quality of 39.77dB for everyone in the coverage area (16 parity packets resulting in a video rate of 1455kbps).

- In order to extend the coverage range, we need to lower the transmission rate, with correspondingly lower PSNR.

The indoor experiments were conducted in the Dibner Building of Polytechnic University. We chose different locations as depicted in Figure 5 in order to obtain variations in PER. The PER results are illustrated in Figure 4(a). Note that, reflections and obstacles play an important role for the indoor experiments and therefore, the increase of the PER indoors is not proportional to the distance between the two stations. For this particular set up with 4 receivers, using 11 Mbps allows a much higher video rate and hence significantly better video quality for all.

Figure 4(b) illustrates the obtained PSNR versus location curve. Comparing the indoor and outdoor experiments, we observe that the higher transmission rate provides higher useful rate and hence higher PSNR, as in the outdoor case. However, due to the relatively small increase in PER as the location changes, the PSNR stays almost constant for different locations at the same transmission rate.

VI. CONCLUSION

In this paper, we explore the dynamics of Forward Error Correction (FEC) schemes in multi-rate wireless local area networks. First, we studied the fundamental behavior of a 802.11b network under the broadcast mode in a real environment, both indoors and outdoors. Then we explore the interaction between PER and FEC for different transmission rates in a real environment. In order to evaluate the system quantitatively, we implemented a prototype using open source drivers and socket programming, and ran experiments. Based on the results of these experiments, we argue that using a higher transmission rate together with stronger FEC is more efficient than using a lower transmission rate with weaker FEC for video multicast.

REFERENCES

[1] A. Basalamah , H. Sugimoto and T. Sato, “Rate adaptive reliable multicast MAC protocol for WLANs,” in *Proceedings of IEEE VTC*, 2006.
[2] C. Huang,J. H. David and C. Chang, “Congestion and Error Control for Layered Scalable Video Multicast over WiMAX,” in *Proceedings of IEEE Mobile WiMAX Symposium*, 2007.
[3] I. Bajic, “Efficient Error Control for Wireless Video Multicast,” in *Proceedings of IEEE MMSP*, 2006.
[4] H. Liu, M. Wu, D. Li, S. Mathur, K. Ramaswamy, L. Han and D. Raychaudhuri, “A Staggered FEC System for Seamless Handoff in Wireless LANs: Implementation Experience and Experimental Study,” in *Proceedings of IEEE International Symposium on Multimedia*, 2007.
[5] J. Villalon, P. Cuenca, L. O. Barbosa, Y. Seok, and T. Turletti, “Cross-Layer Architecture for Adaptive Video Multicast Streaming Over Multirate Wireless LANs,” *IEEE Transactions on Selected Areas in Communications*, vol. 25, no. 4, May 2007. pp 699-711.
[6] P. K. McKinley and A. P. Mani, “An experimental study of adaptive forward error correction for wireless collaborative computing”, in *Proceedings of IEEE SAINT*, 2001.
[7] P. K. McKinley and A. P. Mani, “Experimental Evaluation of Error Control for Video Multicast”, in *Proceedings of IEEE DCSW*, 2001.
[8] L. Rizzo, “Effective erasure codes for reliable computer communication protocols”, in *ACM Computer Communication Review*, April 1997.
[9] A. J. McAuley, “Reliable broadband communications using burst erasure correcting code”, in *Proceedings of ACM SIGCOMM*, September 1990.
[10] “MadWifi: Linux kernel drivers for Wireless LAN devices,” http://madwifi.org/
[11] “Atheros: Chipsets for Wireless LAN devices,” http://www.atheros.com/
[12] “Iperf: The TCP/UDP Bandwidth Measurement Tool,” http://dast.nlanr.net/ Projects/Iperf/.
[13] Joint Scalable Video Model (JSVM), JSVM Software, Joint Video Team, Doc. JVT-X203, Geneva, Switzerland, June 2007

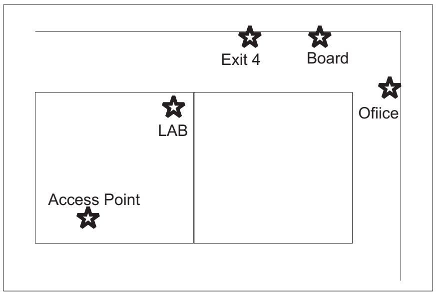


Fig. 5. Indoor locations map