ZigBee-Assisted WiFi Transmission for Multi-interface Mobile Devices

Hua Qin, Yanfei Wang, and Wensheng Zhang

Iowa State University, USA

Abstract. It has been common for a mobile device to have WiFi and Bluetooth interfaces. As the ZigBee technology becomes more mature, it will not be surprising to see the ZigBee interface commonly embedded in mobile devices together with WiFi and other interfaces in the near future. To leverage the ZigBee interface for improving the communication performance of a mobile device, we propose a ZigBee-assisted WiFi transmission system where the ZigBee is used to coordinate the communication activities of WiFi to reduce contention and collision. A prototype of the proposed system and a detailed simulator of it have been implemented; extensive experiments and simulations have been conducted. The results show that, the proposed system can achieve significantly higher throughput and energy efficiency than the IEEE 802.11 protocol.

Keywords: ZigBee, WiFi, IEEE 802.11, throughput.

1 Introduction

Recently, mobile devices are increasingly equipped with multiple network interfaces [1–3]. It has been common for a mobile device, such as smart phone, PDA and laptop, to have both WiFi and Bluetooth interfaces. As the ZigBee technology becomes more and more mature, embedded ZigBee interfaces have emerged and the size is becoming smaller and smaller [4, 5]. It will not be surprising to see the ZigBee interface commonly embedded in mobile devices together with WiFi and Bluetooth interfaces in the near future. With ZigBee interfaces, mobile devices can communicate with various electrical and electronic appliances to realize the smart home entertainment and control, home awareness, mobile services, commercial building and smart industrial plants [6].

The WiFi interface perhaps is the most common interface found in mobile devices for data transfer as it provides good combination of throughout, range and power efficiency. However, the WiFi interface may have to consume a large amount of bandwidth and energy for contention and combating collision, especially when mobile devices located in a small area (e.g., conference room, library, stadium, etc.) all have heavy traffic to transmit. To reduce contention, many protocols have been proposed. However, most of them (e.g., Overlay MAC [7], TDM MAC [8], token-passing MAC [9], etc.) require to either modify the underlying MAC protocol or introduce extra control overhead.

A. Puiatti et al. (Eds.): MobiQuitous 2011, LNICST 104, pp. 248–259, 2012.

The co-existence of the ZigBee and the WiFi interfaces in the same mobile device inspires us to develop new techniques to address the above issue. The key idea is that nearby mobile devices use their ZigBee interfaces to coordinate their communication activities to reduce contention and collision. The rationales behind the idea are as follows. The ZigBee interface and the WiFi interface can use different channels, and hence the coordination using ZigBee interfaces will not consume the WiFi bandwidth. As the WiFi transmission has higher rate and energy consumption than ZigBee transmission, the utilization of WiFi for large-size data transmission and ZigBee for small-size control message transmission presents an ideal, efficient resource allocation pattern. Such collaboration is possible because ZigBee may not be used frequently in the places, such as conference room, library and stadium, where WiFi traffic could be very heavy.

In this paper, we propose a simple yet effective ZigBee-assisted WiFi transmission system for the *high traffic density* scenario. In this system, mobile devices leverage ZigBee communication to form clusters where each cluster has a cluster head and multiple cluster members that can directly communicate with the head via the ZigBee interface. According to the communication demands of individual mobile devices, members in the same cluster collaboratively run a TDMA-like protocol with the ideal goal that, at any moment only one of them attempts to use the WiFi channel so as to eliminate or greatly reduce the contention within a cluster and thus mitigate the contention in the whole network.

The rest of the paper is organized as follows. Section 2 presents the system model. Section 3 elaborates our proposed design. The results of comprehensive simulation and prototype implementation are reported in Section 4 and 5, respectively. Section 6 summarizes related work, and finally Section 7 concludes the paper.

2 System Model

To run our proposed system, each network node (e.g., laptop) has two wireless interfaces: ZigBee (IEEE 802.15.4) and WiFi (IEEE 802.11). We call such nodes **Z-WiFi** nodes. The WiFi interface is for data transmission while the ZigBee interface is for coordinating node transmission activities. Due to current popularity of the IEEE 802.11 protocol, Z-WiFi nodes may co-exist with the nodes that do not have or use ZigBee but use the **S**tandard IEEE 802.11 protocol. We call such nodes **S-WiFi** nodes.

Our design targets mainly at the scenarios where data traffic is heavy due to high node density and/or high packet transmission rate per node. The design objectives are as follows.

- *High Throughput*: By using the information gathered by ZigBee interfaces to carefully schedule the data transmission of WiFi interfaces, our design should reduce the contention among nodes and thereby increase the throughput.
- *Energy Efficiency*: Through reducing the contention experienced by the WiFi interfaces, our design should also decrease the power consumption of nodes.

- *Compatibility*: On one hand, our system should not demand changes in the existing WiFi and ZigBee standards. On the other hand, Z-WiFi and S-WiFi nodes should not harm each other, but should be in the win-win status when co-exist.
- *Fairness*: Our design should organize data transmission of WiFi interfaces in a way that the shared channel is shared relatively fairly among all nodes.

3 Proposed Design

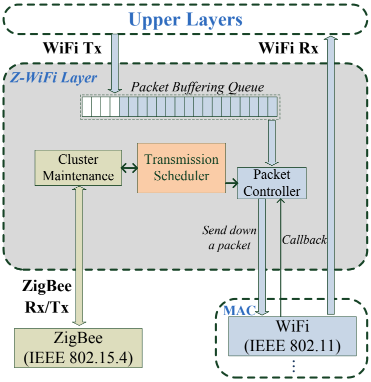


Fig. 1. System architecture

Fig. 1 depicts the architecture of our proposed Z-WiFi system, which is built atop WiFi and ZigBee. Thus, it is transparent to and independent of these standards. The *cluster maintenance* component works through communication over the ZigBee interface. A *packet buffering queue* is used to temporarily buffer packets from the upper layer. Through monitoring the status of the queue, packet arrival rate can be inferred, based on which the *transmission scheduler* dynamically computes the TDMA-like schedule for WiFi transmission within a cluster. The schedule is executed by the *packet controller* component which controls the timing and speed for passing packets in the *packet buffering queue* down to the underlying IEEE 802.11 MAC layer.

In the section, we present the deign details of our proposed Z-WiFi network. Briefly, we first present the cluster formation scheme. Then, the intra-cluster and the inter-cluster coordination are elaborated, respectively. After that, heuristics is designed to deal with practical issues.

3.1 Cluster Formation

To facilitate the coordination of WiFi transmission for reduced contention, we propose to organize nodes that have potential need for contention into a single cluster through ZigBee communication. Based on existing cluster formation protocols [13], we propose a cluster formation scheme efficient for the scheduling of WiFi transmission.

Initially, each node marks itself as a free node (denoted as **FN**). To obtain information about neighboring nodes, each node periodically broadcasts a *beacon message*, defined as $\langle Node\_id, CH\_id, i, r_i \rangle$, via its ZigBee interface. Here, $Node\_id$ is the network-wise unique id of the sender, CH_id is the node id of its cluster head (denoted as **CH**) if the sender has joined a cluster (otherwise it is empty), and i is a cluster-wide unique *index* of the sender, assigned by the corresponding CH, when it joins the cluster. Besides, r_i is its current packet arrival rate (in the unit of *bits/second*) of the node with index i, estimated through monitoring the status of its packet buffering queue. Note that, if the sender is a cluster member (denoted as **CM**) or a FN, r_i is the packet arrival rate of its own; if it is a CH, r_i is the sum of packet rates of all nodes in its cluster. The usage of r_i and i is to be detailed later.

Based on beacon exchange, each node can maintain a neighbor information list to record the most recent information about its neighbors. If a FN has heard a beacon from one or multiple CHs, it chooses the one whose cluster has the *smallest* packet arrival rate to join. Otherwise, if a FN does not find any CH after a certain rounds of beacon exchange, it announces itself as a CH candidate by broadcasting a *formation* packet piggybacking the number of FNs in its neighborhood. When a node that is not a CH candidate first receives the formation packet, it waits for a certain period of time to overhear other possible formation packets; when the backoff expires, the candidate CH having the largest number of FNs is chosen as its CH and a *registration* packet is sent back to the candidate to join. Upon receiving a registration packet, the candidate node becomes a new CH. In response to each registration from a new CM, the CH sends back an *index* packet, in which a cluster-wide unique index i (i is a positive integer) is assigned to the CM. Note that, the index of a CH is 0.

3.2 Intra-cluster Coordination for WiFi Transmission

Based on the cluster structure, WiFi transmissions of nodes within the same cluster between CH and CMs for reduced contention are coordinated. Each CM is time-synchronized with its CH.

Besides, each node measures the packet arrival rate (i.e., r_i) at its packet buffering queue, rather than at application layer. When packet buffering queue is full, any incoming packet from upper layer is dropped, which imposes a limit on the value of r_i. Hence, r_i cannot be infinitely large.

With the synchronized time reference, time is divided into frames and each frame is further sliced into slots of equal length. The length of a slot, denoted as τ_w, is the empirical time needed to send a packet through WiFi interface. The CH assigns the slots in each frame to the nodes in its cluster, according to their packet arrival rates. In the following, we show how the CH computes the WiFi transmission schedule (i.e., the slots to transmit), how it is represented and how the CH updates the schedule to its CMs by using the ZigBee interfaces.

A WiFi transmission schedule is represented and sent as a sequence of binary bits, which can be contained in the payload of a single ZigBee packet. A sequence consists of many sub-sequences of 0(s) separated by a 1. For example, sequence

$$\underline{00000}11\underline{0000}1\underline{000}1\underline{00}1\underline{000}100\cdots0$$

represents that a WiFi transmission schedule, where each frame has 17 slots, nodes with indices 0, 1, 2, 3, 4 and 5 are assigned with 5, 0, 4, 3, 2 and 3 slots, respectively. Node 0 (i.e., the CH) can perform WiFi transmission during the first 5 slots of each frame, node 1 may not exist or has no packet to send, node

2 can perform WiFi transmission during the 6th to the 9th slot of each frame, and so on and so forth. WiFi transmission schedule periodically is updated and broadcasted by the CH via its ZigBee interface as the packet arrival rate may change in each node.

Particularly, in our experiments, we set the payload size to 28 bytes, which is the default payload size used by TinyOS. Once the payload size is determined, the maximum number of slots in a frame is also determined. We denote the maximum number of slots in a frame as f_w^{max}. Also, we use r_i to denote the packet rate of node with index i ($i = 0, \cdots, N-1$) in the cluster, recalling that each node is assigned a unique index. Let δ ($0 < \delta \leq 1$) be a predetermined system parameter. The number of slots allocated to each node i (denoted as n_i) and the actual number of slots composing a frame (denoted as f_w) is computed as follows:

$$n_i = \left\lfloor \min \left\{ \delta \cdot \frac{r_i}{B \cdot \tau_w}, f_w^{max} \cdot \frac{r_i}{\sum_{j=0}^{N-1} r_j} \right\} \right\rfloor > 0, \tag{1}$$

$$f_w = \sum_{i=0}^{N-1} n_i \leq f_w^{max}, \tag{2}$$

where B is the WiFi bandwidth. Thus, $r_i/B\tau_w$ represents the expected number of packets sent by node i. The rationale behind the slot computation is of three folds:

- For the sake of fairness, the number of slots allocated to a node is proportional to the packet arrival rate of the node while the total number of slots composing a frame should not exceed f_w^{max}.
- The ratio between the number of slots and the packet arrival rate is determined by system parameter δ. The larger is δ, the longer is a frame and the larger number of consecutive slots a node can use for WiFi transmission, and vice versa. Through our experiments, increasing δ leads to decrease in energy consumption and increase in packet delay, and vice versa. To balance energy consumption, δ is set to 0.2.
- Based on Eq. (1), the *clustering condition* can be defined as follows: *a FN node can join or form a cluster only if for any node i (including itself) in the resulted cluster* $n_i > 0$ *can be satisfied.* On one hand, a node with very few packets to send do not need to join or form a cluster and it can just use the IEEE 802.11 protocol as a FN. On the other hand, a node with a high packet rate should not be allowed to join a cluster if its joining makes any existing node in the cluster have *zero* slot to transmit. Thus, after a certain period of time, it will attempt to form a new cluster.

Ideally, each node transmits data through its WiFi interface only during the slots assigned to it, and one packet uses one slot time (i.e., τ_w) to be transmitted. It follows that n_i packets should be sent down to the underlying 802.11 MAC layer in each frame. However, in practice, this is hardly true.

To make full use of each slot, we propose to use the *callback* (i.e., notification of the completion of a packet transmission) from the underlying MAC layer to control the timing for passing packets downwards, as illustrated in Fig. 1. Specifically, when the scheduled transmission time (i.e.,$n_i\tau_w$) begins, the packet buffering queue delivers a packet to the MAC layer. As long as the scheduled time does not run out and there is an available packet for transmission, a packet will be pushed down to the MAC layer once the callback of previous packet is received.

3.3 Inter-cluster Dynamics for Dealing with Mobility

Due to mobility, a CM may move out the range of its CH and join another cluster; a FN may discover a CH and join the cluster headed by that CH; a CH may move into the range of another CH and their clusters may be merged to reduce the number of co-existing clusters and hence inter-cluster contention.

Cluster Switching. When a CM with index i finds it has moved out of the ZigBee communication range of its CH, i.e., failing to receive beacon from its CH for a certain time, it attempts to discover nearby CHs by overhearing beacons. If it finds some CHs, it joins the cluster that has the lowest overall packet arrival rate. If no CH is found in vicinity, it becomes a FN, which can either join another cluster, or form its own cluster. Note that, if a CH fails or is turned off, its CMs will not be able to receive beacon messages from it, in which case they will react as if they have moved out of the communication range of the CH and perform cluster switching as depicted above.

Cluster Joining. When a CM or CH becomes a FN, it first tries to join other cluster by turning on its ZigBee and listening for a certain time. If it finds some CHs in the vicinity, a registration packet is sent. Upon receiving the registration packet, the CH acknowledges that node by replying an *index packet* containing a unique index (typically the *smallest unused* index in the cluster) assigned to that node, if the clustering condition (See Eq. (1)) can be satisfied. Once the index packet is successfully received by the FN, it becomes a CM of that cluster. If no CH is found, it starts the cluster formation process as described in Section 3.1, if the clustering condition can be satisfied.

Cluster Merging. To dynamically minimize the cluster density and hence reduce inter-cluster contention, cluster merging is proposed as follows. As CHs are always awake, they may overhear WiFi transmission schedule packets from nearby clusters. When a CH (CH1) overhears a schedule packet from another CH (CH2), it checks if it can cover more than half of CMs of CH2. If so, merging process will be conducted through the negotiation between these two CHs. As a results, the nodes that are in the cluster of CH2 and covered by CH1 are merged into the cluster of CH1, while the rest of CMs become FNs, which with either join other clusters or form a new cluster later.

3.4 Practical Issues

Turning on/off ZigBee Our system is designed mainly to improve WiFi performance in high-contention scenarios, and the IEEE 802.11 protocol can already achieve the optimal throughput when the contention is low. To avoid unnecessary control overhead, we propose a simple heuristic parameter γ for turning off ZigBee interfaces of Z-WiFi nodes when the contention is low and turning on them when the contention is high. The nodes without using ZigBee interface run the IEEE 802.11 protocol.

Specifically, each node records transmission time (i.e., duration from the arrival of a packet to the reception of corresponding ACK) of the most recent outgoing packets. Let T_{pkt} be average transmission time, then

- ZigBee is turned off, if $T_{pkt} < 0.5 \times \gamma\tau_w$;
- ZigBee is turned on, if $T_{pkt} > 1.5 \times \gamma\tau_w$.

$\gamma\tau_w$ represents the expected packet delivery delay when system throughput is saturated. The selection of γ is to be studied in Section 4.1.

Co-existence of Z-WiFi and S-WiFi. In practice, Z-WiFi and S-WiFi nodes may co-exist in a small area. Since they run different protocols for data transmission, the resulting performance is different. Generally, S-WiFi nodes can achieve better performance than Z-WiFi nodes, because S-WiFi nodes can contend for channel occupation all the time while Z-WiFi nodes are only allowed to access the channel within their scheduled time slots. To address this practical issue, we propose to dynamically tune the contention window of Z-WiFi nodes so as to achieve a win-win status, in which both types of node can achieve throughput improvement and good fairness. Due to space limit, the detailed design is presented in [20].

4 Simulation

To evaluate our proposed system in a large-scale network, we simulate the system with ns2 simulator. In the simulation, the following major metrics are studied:

- *Network throughput* (Mb/s) is the total amount of data successfully transmitted (i.e., ACKed at sender side) in the network. To measure the throughput, each node runs an application which keeps sending UDP packets and by default totally all these nodes generate the data input with an average rate of 20.4Mb/s (i.e., 22 packets/s at each node on average). All the packets have maximum payload size.
- *Energy consumption* (J/Mb) is computed as the total amount of energy consumed by all network interfaces of all nodes divided by the number of Mbs of data that has been successfully transmitted. The energy consumed by the WiFi interface is measured according to the specified power consumption rate of SX-SDWAG 802.11g wireless module [16] (i.e., 1047mW for transmission, 513mW for reception and 420mW for being idle) and the power consumed by

the ZigBee interface is measured according to the specified power consumption of CC2420 RF transceiver [17] (i.e., 52.2mW for transmission, 56.4mW for reception, 1.28mW for being idle, 0.06μW for sleeping and 0.06mW for transition).

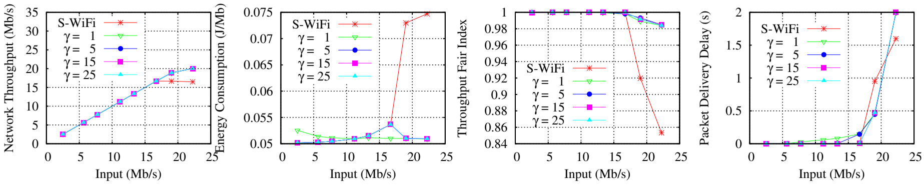
- *Throughput fairness* is measured with respect to the fairness index (FI) [18], which is defined as $FI_{tp} = \frac{\mu(\chi)}{\mu(\chi)+\sigma(\chi)}$, where $\mu(\chi)$ and $\sigma(\chi)$ are the mean and the standard deviation of χ at all network nodes. χ is the ratio of throughput to input. Obvious, FI_{tp} is between 0 and 1. The more closer FI_{tp} approaches 1, the better is the fairness.

Unless otherwise specified, our simulation use the settings shown in the table below. Also, we adopt the random waypoint mobility model, where the pause time is fixed to 20s and the maximum speed is 2m/s. Besides collision-caused drops, each node intentionally drops 2% incoming packets on ZigBee communication to simulate the packet loss due to interference from WiFi. The default IEEE 802.11g protocol is used.

Number of nodes	50	Network scale	100m × 100m
Range of WiFi (R_w)	120m	Range of ZigBee (R_z)	60m
Simulation time	1 hour	WiFi slot length (τ_w)	0.001s
ZigBee on/off parameter (γ)	15	Packet buffer size	50 packets

4.1 Comparing with S-WiFi System and Studying Parameter γ

To find the best time to turn on ZigBee so as to maximize the performance, we compare Z-WiFi system, configured with four different values of γ (i.e., 1, 5, 15 and 25), with S-WiFi system.



(a) (b) (c) (d)

Fig. 2. Choosing parameter γ by comparing with S-WiFi

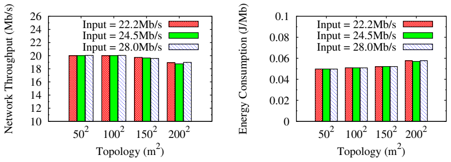
From Fig. 2a, we can see that when network input is below 17Mb/s, S-WiFi system can almost deliver all incoming packets. When input is beyond 17Mb/s, S-WiFi nodes reach the maximum throughput. At this time, ZigBee interface of Z-WiFi nodes should be turned on to assist WiFi transmission. As shown in Fig. 2a and 2c, $\gamma = 5, 15$ or 25 can precisely render ZigBee turned on at the right time. This is because, due to accumulated waiting delay in the packet buffer queue, packet transmission delay rises up drastically (from less than one millisecond to more than hundreds of milliseconds) once S-WiFi system gets saturated. Thus, large values of γ (e.g., $\gamma > 5$) can work appropriately. Particularly, when ZigBee interface is turned on (i.e., input exceeds 17Mb/s), energy consumption drops rapidly, as shown in Fig. 2b, which shows that our proposed system can save energy.

When $\gamma = 1$, ZigBee interface is turned on when network input (i.e., contention) is low. At this time, our protocol cannot help, as the S-WiFi system has already achieve the optimal throughput. Hence, the overhead introduced for ZigBee communication makes Z-WiFi systems consume more energy.

In addition, we also measure average packet delivery delay from application layer, as illustrated in Fig. 2d. From the results, setting γ to 15 or 25 can guarantee that Z-WiFi system can achieve no longer packet delivery delay than S-WiFi system when input is below 21Mb/s. When input is above 21Mb/s, our system also becomes saturated and thereby packet delivery delay increases. Note that the packet delivery delay of Z-WiFi system is longer than that of S-WiFi system only when the throughput of Z-WiFi is higher than S-WiFi.

To summarize from the above results, *our proposed system can improve the network throughput by 18%, reduce the energy consumption by 32% and provide much better fairness, when the network traffic density is high.*

4.2 Performance with Different Network Scale



(a) (b)

Fig. 3. Impact of network scale on performance

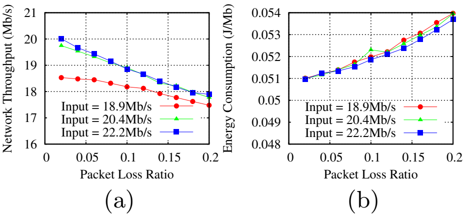
Fig. 3 shows how our system works with different network scale. Generally, the throughput slightly decreases as the scale of the network becomes larger, due to the number of clusters increasing. When the number of clusters within WiFi transmission range increases, contention gets more severe, which degrades the performance. However, the number of clusters will not become too large, since cluster merging mechanism is applied, which can ensure the number of interfering clusters close to $\lceil R_w^2/R_z^2 \rceil$ (e.g. 4 under our simulation). For energy consumption illustrated in 3b, more clusters consume more energy in transmission coordination and cluster maintenance.

4.3 Impact of ZigBee Packet Loss on Performance

Apart from random collision-caused packet loss, we also study the packet loss due to other environmental phenomena (e.g., interference, obstacle, multipath, etc.). Thus, we conduct a simulation by varying packet loss ratio from 2% to 20%. As shown in Fig. 4, our performance degrades slightly as loss ratio gets larger. For throughput, it is because of the insufficient utilization of channel caused by increasing delay or error in updating WiFi transmission schedule. The energy consumption increases mainly because of the increased energy consumption for contention caused by schedule inconsistencies, resulted from packet loss.

5 Implementation

5.1 Prototyping



(a) (b)

Fig. 4. Impact of ZigBee packet loss on performance

As a proof of concept, we implement a prototype of our proposed system. We build a testbed with 10 DELL D-Series laptops (called *nodes* hereafter), each running the Ubuntu Linux 8.10 (kernel 2.6.27-17-generic). Each node is also equipped with a D-Link WNA-2330 Wireless G Notebook Adapter (108Mbps, 802.11g, Atheros chipset, PCMCIA) and a Crossbow telosB mote (i.e., ZigBee interface). Note that *the wireless adapter is built with the state-of-the-art technology, which can deliver higher throughput than standard 802.11g devices.* The scheduling of WiFi transmission is implemented upon MadWiFi [19], an open-source driver for Atheros chipset-based 802.11 Wireless LAN devices. The prototyped ZigBee communication is implemented upon TinyOS 2.1.1 platform, where 10 nodes form a cluster. The WiFi interfaces of all nodes run in the ad hoc model and are tuned to Channel 3, and the ZigBee interfaces are tuned to Channel 26; thus, the interference between them is small. Besides, the implementation of transmission scheduling is based on *software timer* provided by Linux kernel, which can allow a minimum granularity of 1μs.

Experiments have been conducted on the prototyped system to evaluate the feasibility and the performance of our designed system. For comparison, two sets of experiments are conducted by running the IEEE 802.11 protocol and our proposed system, respectively. Through the experiments, we measure the maximum network throughput as the number of nodes increases from 2 to 10. To measure the maximum throughput, each node generates UDP traffic of 34.8 Mb/s. Each packet has a payload of 1450 bytes, which makes the overall packet to exactly fit into a single MAC-layer frame. The duration of each experiment run is 5 minutes. The experiment is conducted three times. Besides, $n_i = 10$ and $\tau_w = 0.001$s.

5.2 Experiment Results

The experiment results are shown in Fig. 5. In general, compared with the IEEE 802.11 protocol, our proposed system can improve the network throughput significantly. Particularly, when the number of involved nodes reaches 10, the improvement of throughput can be as high as 49.1%. As expected, our proposed system outperforms the IEEE 802.11 protocol when the number of transmitters is large (e.g., more than 4 nodes in our experiment). As that number keeps increasing, the difference becomes more significant because the IEEE 802.11 protocol suffers from severe contention and the throughput drops fast.

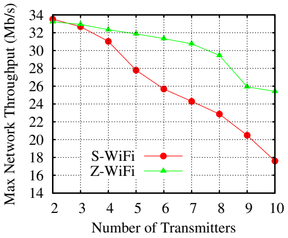


Fig. 5. Maximum Network Throughput

Moreover, the standard deviation (STDV) of throughput among different nodes is also measured, as shown in the table below. From the results, we can see that using our proposed system introduces much lower throughput STDV, which indicates better throughput fairness.

# of transmitters	Throughput STDV of S-WiFi	Throughput STDV of Z-WiFi
4	1.1016	0.1780
6	0.8016	0.1281
8	0.7698	0.1775

Through the experiments, our proposed system has been shown to be able to improve throughput significantly and provide fair sharing of bandwidth.

6 Related Work

Recently, some research has conducted to investigate co-located interfaces for improving the performance of IEEE 802.11 network. One of the first work is Blue-Fi [1], which brings forth the idea of using other co-located interface to assist WiFi transmission. It uses the co-located Bluetooth to predict the availability of the WiFi connectivity by using user's trend of repeatedly encountering the same set of bluetooth devices and cell-towers. Different from Blue-Fi, our system uses ZigBee interface, which has a much longer communication range. Thus, it can provide a better communication capability under the mobile environment. Our proposed system is motivated by this feature. Because of using different hardware and methodologies, the accomplishment of Blue-Fi and Z-WiFi are also different. Besides, ZiFi [2] utilizes ZigBee radios to identify the existence of WiFi networks through WiFi beacons, while WiZi-Cloud protocols [3] have been proposed to use WiFi-ZigBee radios on mobile phones and Access Points to achieve ubiquitous connectivity, high energy efficiency, real time intra-device/inter-AP handover. Unlike those work, our work focuses on improving the performance WiFi transmission under the DCF through reducing contention. In general, the previous work targets on saving energy, but our work aims to improve the throughput, power efficiency and fairness.

7 Conclusion

In this paper, we have proposed a simple yet effective system for ZigBee-assisted WiFi transmission. Mobile devices form clusters. Coordinated through ZigBee interfaces, members in each cluster take turns to transmit, resulting in reduced contention and collision. Results of experiment and simulation have verified our design by showing that, the throughput, power consumption and fairness can be improved.

Acknowledgments. This work is supported partly by the NSF under Grants CNS-0834593 and CNS-0831874, and by the ONR under Grant N00014-09-1-0748.

References

1. Ananthanarayanan, G., Stoica, I.: Blue-Fi: Enhancing Wi-Fi performance using Bluetooth signals. In: MobiSys (2009)
2. Zhou, R., Xiong, Y., Xing, G., Sun, L., Ma, J.: ZiFi: Wireless LAN Discovery via ZigBee Interference Signatures. In: MobiCom (2010)
3. Jin, T., Noubir, G., Sheng, B.: WiZi-Cloud: Application-transparent Dual ZigBee-WiFi Radios for Low Power Internet Access. In: InfoCom (2011)
4. One RF Technology, Short Range RF, `http://www.one-rf.com`
5. LS Research, LLC., ZigBee ProFLEX Module, `http://www.lsr.com`
6. Wikimedia, ZigBee, `http://www.wikipedia.org/wiki/Zigbee`
7. Rao, A., Stoica, I.: An overlay MAC layer for 802.11 networks. In: MobiSys (2005)
8. Koutsonikolas, D., Salonidis, T., et al.: TDM MAC Protocol Design and Implementation for Wireless Mesh Networks. In: CoNEXT (2008)
9. Liu, I., Takawira, F., Xu, H.: A Hybrid Token-CDMA MAC Protocol for Wireless Ad Hoc Networks. In: TMC (2008)
10. Kim, S.M., Chong, J., Jung, C., et al.: Experiments on Interference and Coexistence between Zigbee and WLAN Devices Operating in the 2.4GHz ISM band. In: NGPC (2005)
11. Shuaib, K., Boulmalf, M., Sallabi, F., Lakas, A.: Co-existence of Zigbee and WLAN: A performance study. In: Proc. IEEE/IFIP Int. Conf. Wireless & Optical Communications Networks (2006)
12. Wikimedia, IEEE 802.11, `http://en.wikipedia.org/wiki/802.11`
13. Wan, J., Yuan, D., Xu, X.: A review of cluster formation mechanism for clustering routing protocols. In: ICCT (2008)
14. Wu, H., Peng, Y., Long, K., Cheng, S., Ma, J.: Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis And Enhancement. In: InfoCom (2002)
15. Ni, Q., Li, T., Turletti, T., Xiao, Y.: Saturation throughput analysis of error-prone 802.11 wireless. In: JWCMC (2005)
16. Silex Wireless Modules, `http://www.silexamerica.com`
17. CC2420 RF transceiver, `http://www.chipcon.com`
18. Qiao, D., Shin, K.G.: Achieving Efficient Channel Utilization and Weighted Fairness for Data Communications in IEEE 802.11 WLAN under the DCF. In: IWQoS (2002)
19. MadWifi, `http://madwifi-project.org`
20. Qin, H., Wang, Y., Zhang, W.: ZigBee-Assisted WiFi Transmission for Multi-Interface Mobile Devices. Technical Report (2011), `http://www.cs.iastate.edu/~qinhua/papers/zwifi.pdf`