Peer-to-Peer Netw. Appl.
https://doi.org/10.1007/s12083-017-0619-8

Cooperative channel allocation and scheduling in multi-interface wireless mesh networks

Xiaoheng Deng[1] · Jie Luo[1] · Lifang He[1] · Qiang Liu[1] · Xu Li[1] · Lin Cai[2]

Received: 13 February 2017 / Accepted: 31 October 2017

Abstract Cooperative channel allocation and scheduling are key issues in wireless mesh networks with multiple interfaces and multiple channels. In this paper, we propose a load balance link layer protocol (LBLP) aiming to cooperatively manage the interfaces and channels to improve network throughput. In LBLP, an interface can work in a sending or receiving mode. For the receiving interfaces, the channel assignment is proposed considering the number, position and status of the interfaces, and a task allocation algorithm based on the Huffman tree is developed to minimize the mutual interference. A dynamic link scheduling algorithm is designed for the sending interfaces, making the tradeoff between the end-to-end delay and the interface utilization. A portion of the interfaces can adjust their modes for load balancing according to the link status and the interface load. Simulation results show that the proposed LBLP can work with the existing routing protocols to improve the network throughput substantially and balance the load even when the switching delay is large.

Keywords Multi-channel multi-interface · Wireless mesh network cooperative networking · Link layer protocol · Channel assignment · Load balance

Preliminary results of this work have been presented in IEEE HPCC'13 [24]. The new contributions of this manuscript included a more detailed framework design in Section 3 (Fig. 1) and the analysis of the channel assignment by Eqs. 1–2, a new receiving interface task allocation method based on Huffman tree in Section 4.1.2, a full consideration of the switching delay in Eq. 5, the detailed analysis on the process of interface modes switching in Section 4.3, and a more comprehensive protocol performance evaluation by simulation in various scenarios with different protocols (as shown in Figs. 4, 5, 6).

✉ Lin Cai
cai@ece.uvic.ca

Xiaoheng Deng
dxh@csu.edu.cn

Jie Luo
164611034@csu.edu.cn

Lifang He
helifang@csu.edu.cn

Qiang Liu
jasnykra@csu.edu.cn

Xu Li
lishlxg@csu.edu.cn

1 School of Information Science & Engineering, Central South University, Changsha 410083, China

2 Department of Electrical & Computer Engineering, University of Victoria, Victoria, BC, Canada

1 Introduction

Wireless mesh networks with multiple interfaces and multiple channels have attracted great attention recently. In the IEEE 802.11 a and b/g standards, there are 12 and 3 non-overlapping channels respectively. Thus, in a multi-channel network, non-overlapping channels can be used simultaneously and the network has the potential to achieve a much higher throughput. Furthermore, the reduced hardware cost makes it possible to equip a node with multiple interfaces, so each node can use multiple non-overlapping channels simultaneously. Considering the mutual interference and load balance issues, how to cooperatively assign channels and schedule links are two key issues for multi-channel and multi-interface wireless mesh networks, which motivates this work.

Published online: 13 November 2017

In the multi-interface, multi-channel mesh networks, the number of the channels may be larger than that of the interfaces of a node. To maximize the network throughput, an optimal channel assignment should allow the maximum number of concurrent transmissions with bounded mutual interference, which has been proved to be NP-hard [1, 26]. Many heuristic algorithms have been proposed to minimize the interference, e.g., BFS-CA [2], LA-CA [1] and PCU-CA [3].

On the other hand, channel assignment may affect network connectivity [12, 13, 31]. There are two main approaches to address it, the dedicated channel assignment [3] and the synchronous channel assignment [4]. The later requires time synchronization and its performance may degrade when the transmission error rate is high, as the retransmission opportunities are limited. So, the dedicated channel assignment is adopted to ensure the connectivity in this paper.

Given that nodes have multiple interfaces, how to balance the load through cooperative channel allocation is critical and it is the main focus of this work. To address this issue, we propose a load balance link protocol (LBLP) by channel allocation and link scheduling to improve the network throughput and balance the traffic load. The main novelty and contributions of our approach are as follows. The interfaces are divided into three categories, static, dynamic, and adaptive ones. First, for the receiving interfaces, a channel assignment estimating the interference accurately and a load allocation algorithm based on Huffman tree are developed to reduce the interference and balance the receiving traffic load. Second, the sending interfaces adopt a dynamic link schedule mechanism which improves the interface utilization and reduces the delay. Third, to balance the load between the receiving and sending interfaces, according to the traffic statistics, the adaptive interfaces can change between the receiving and sending modes automatically. Simulation results show that our protocol can reduce the interference and balance the traffic load to offer a higher network throughput, and LBLP can work well with the existing routing protocols such as AODV and DSR in multi-channel, multi-interface networks.

The rest of the paper is organized as follows. In Section 2, we review the related work. Then, we present the framework of LBLP in Section 3. The channel assignment for static interfaces, the link scheduling for dynamic interfaces and the control of the adaptive interfaces are presented respectively in Section 4. The performance of LBLP is evaluated in Section 5, followed by the concluding remarks in Section 6.

2 Related work

In a multi-interface, multi-channel mesh network, several multi-channel MAC protocols have been proposed [10, 11]. Although they can achieve good performance, they are not compatible with the IEEE 802.11 standards. Here, we focus on the logic link control protocols compatible with the MAC protocol specified in the standards.

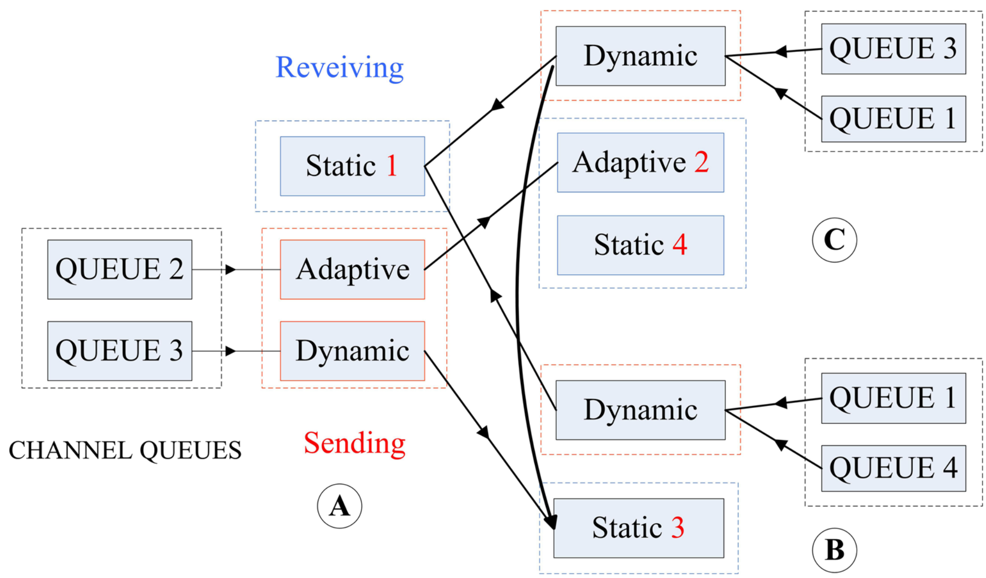
Channel assignment can be classified into centralized and distributed ones [13, 30]. Although centralized algorithms may find close-to-optimal solutions, they rely on global traffic knowledge and reliable communications to deliver the assignment results to all nodes timely. In this work, we focus on the distributed algorithm design. In the network, ideally, every interface should always select a channel free of interference. This is the motivation of AODV-MR [15] and CA-AODV [16]. Interference-free assignment may not always be possible in realistic situations. To address this issue, AODV-MR adjusts the channel assignment dynamically, while frequent switching and time synchronization may reduce the efficiency. To minimize the switching and synchronization overhead, the key is to estimate the interference accurately and select the channel with the minimum interference.

Solutions in [17, 18, 29] use the physical channel characteristics, such as SINR and SNR, to measure the interference. They can obtain the real-time interference considering the path-loss effect as well. If the wireless channel or interference channel vary over time quickly, the result may not be reliable. [3, 19] considered the number of the interfaces using the same channel for load balancing. The complexity is low and it is easy to implement. On top of it, this work further considers the effect of the nodes' distribution and status to fine tune the channel assignment to achieve better load balancing.

For link scheduling, a link layer protocol called SSCH was proposed [4], which is easy to be extended to the multiple-interface cases. However, it requires a small interface-switching delay (in the range of 80μs), while the switching delay of many existing hardware is several milliseconds. Therefore, our goal is to deal with the case when the switching delay is large.

Cooperative optimization problem is a hot topic in recent years [27, 28, 32]. Dong et al. [6] propose an advanced BCMN/A protocol, which broadcasts intra-cluster and returns multi-ACK for each data received from clusters in order to jointly optimize lifetime and transport delay. Experiments shows that BCMN/A can increase the network lifetime by more than 8 percent and reduce network delay by more than 25 percent. Ren et al. [7] propose a channel management scheme JASC, which can jointly manage channel access and sampling rate. They further propose a dynamic channel accessing protocol [8, 34], which can dynamically adjust the transmission power of cluster heads and determine the channel sensing and accessing sequence, the proposed schemes can significantly reduce the energy consumption of data transmission and outperform the existing

Fig. 1 Framework of LBLP



work. However, they ignored the consumption of channel sensing and switching.

Considering the characteristics of the mesh traffic, Mogaibel et al. proposed a multi-radio on demand distance vector routing protocol (AODV-MRCR) based on the AODV-MR protocol in [20]. AODV-MRCR reserves a unique list of channels for the gateway traffic and handles the gateway traffic and local traffic differently. Gateway traffic has a higher priority, so that several channels are reserved for it while only one fixed channel is used for local traffic, which may not be desirable when the local traffic volume is high. A more general load balance solution adaptive to local and gateway traffic in mesh networks is desirable to further improve the efficiency. Thus, we propose a load balance link protocol (LBLP) focuses on the channel assignment and the link scheduling. LBLP can fully utilize all interfaces by a Huffman tree-based channel assignment and task allocation algorithm. Meanwhile, it balances the traffic by adapting interfaces considering both the gateway traffic and local traffic. Consequently, LBLP reduces the average queuing delay and maximizes the network throughput.

3 System model

We consider a mesh network where the mesh routers are static, and focus on the situation that the load of each router slowly changes over time.[1] A factor that affects the utilization of multiple-interfaces is the switching delay, which is the delay of an interface switching from one channel to another. Typically, the switching delay is about several milliseconds [5, 33], which cannot be ignored. Frequent switches cause low efficiency in channel utilization while a static configuration may lead to undesirable performance due to traffic dynamics. We need to consider the tradeoff.

We first present the framework of the proposed link protocol, LBLP, as shown in Fig. 1. There are three multi-interface mesh nodes, A, B, and C. For a multi-interface node, each interface can work in one of the two modes, the receiving mode and the sending mode.

To ensure the connectivity, one interface is always in the receiving mode, called the static interface (whose working channel does not change), and another one is always in the sending mode, called the dynamic interface (whose working channel changes over time), respectively. Other interfaces, called adaptive ones, can adjust the mode to balance the load among channels and interfaces.

For an interface in the receiving mode, it works in a dedicated channel. After choosing the working channel, the channel information will be broadcasted on all available channels to the neighbors. The neighbors will keep a record of this information to send packets to the neighbors accordingly. The static channel does not change to ensure the connectivity between neighbors. The packets being sent through the same channel will enter the same queue, and the sending interfaces will take turns to send the packets in different queues.

4 Link protocol design and implementation

The design and implementation of LBLP mainly includes three aspects: channel assignment and task allocation for

[1] The proposed protocol can still be applicable for the scenarios that the routers are mobile or the load changes fast with certain degree of performance loss, which is beyond the scope of this paper.

the receiving interfaces, link scheduling for the sending interfaces, and management of the adaptive interfaces.

4.1 Receiving interfaces

For the receiving mode interfaces, we use dedicated channels to reduce the complexity in sender and receiver coordination and ensure the network connectivity, and to reduce the channel switching overheads.

According to the framework of LBLP, the sending interfaces should use the channel according to the neighbors' advertised receiving interfaces. Thus, the key is to carefully determine the receiving interfaces to reduce the mutual interference.

If the receiver has two or more receiving interfaces associated with different channels, the sender can select one to send packets. If one receiving interface handles much more packets than the others, not only the utilization of other interfaces is decreased, but also the throughput may be degraded due to more collisions. The task allocation scheme is mainly designed to handle this issue.

4.1.1 Channel assignment

For channel assignment, the first principle is that the interfaces of the same node try not to select the same channel to avoid mutual interference. Second, ideally, we should select a channel suffering the least interference from others. Thus, it is essential to define an accurate method to measure the channel interference, which is discussed below.

Interference model The interference model is shown in Fig. 2, where R_d, R_{tx}, and R_{cs} in the model represent the distance between the transmitter and the receiver, the transmission range and the carrier sense range (equal to interference range), respectively. Generally speaking, the interference range is larger than the transmission range, so $q = R_{cs}/R_{tx}(q \geq 1)$. We assume that the effect of the concurrent transmissions beyond the interference range on the tagged transmission is negligible.

As shown in Fig. 2, when node A is sending packets to B, node C is interfered by A, and node D is in the interference range of B. Thus, node C and node D should avoid transmitting at the same time as node A, which can eliminate the hidden terminal problem. Node E is out of the interference range of both node A and node B, so nodes E and A can transmit simultaneously, which can avoid the exposed terminal problem. Normally, the number of nodes within the interference range is larger than that of the orthogonal channels, thus it may not be always possible to find a channel that has not been used by any neighbor in its interference range. Next, we try to estimate the interference of a channel.

Active status of channels When a node has to select the same channel as one of its neighbors, it is desirable to select the channel least used. Here, we let each interface monitor the wireless activities to estimate the active status of a channel, and measures the percentage of the busy time on the channel ch by Eq. 1.

$$P(ch) = t_{active}/T \tag{1}$$

where t_{active} is the active time of the channel ch during the period of T.

Interference estimation Considering the scenario shown in Fig. 2, for node B, the interference from node C is typically much higher than that from node D, given the path-loss behavior of wireless propagation. Thus, the distance to the interferer should be considered when estimating the interference and SINR. The further the distance is, the

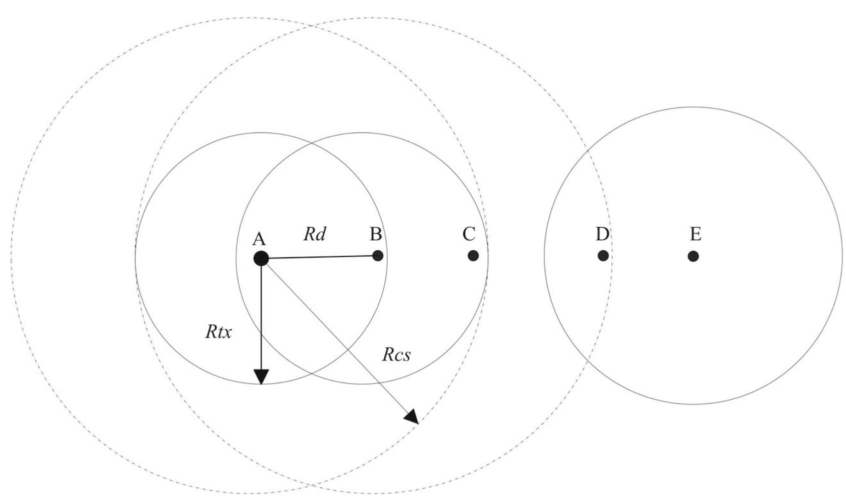


Fig. 2 The interference model

less the interference is. The interference from each interface is estimated to be proportional to $(1/d_i)^\alpha$ where d_i is the distance to interface i, and α is the path-loss exponent taking the value from 2 to 6. For free-space, $\alpha = 2$; for the simplified two-path model, $\alpha = 4$. For indoor and urban radio channels, the path-loss exponent will change with the building and street layouts, as well as with construction materials and density and the height of the buildings in the area. The results of indoor radio propagation studies show that the value of α can be smaller than two for corridors or large open indoor areas, and as high as six for metallic buildings [25].

To combine the factors discussed above, we rank the interference of a channel ch according to Eq. 2.

$$Infer(ch) = \sum\nolimits_{i \in N(ch)} (1/d_i)^\alpha \times P(ch), \tag{2}$$

where i belongs to the set of interferers $N(ch)$ who work on the channel ch and are within the interference range of the tagged node, and $P(ch)$ is the active status defined in Eq. 1. Then, an interface should select the channel suffering the least estimated interference according to Eq. 2.

The channel assignment procedure is listed in Algorithm 1. For each receiving interface, a node selects the channels free of interference as the best choice (Line 12); otherwise, the channel with the least interference is selected (Line 16).

Algorithm 1 Channel assignment

Require: the interfaces set $N(ch)$; $P(ch)$, ch=1,2,...,j, the active status of every neighbor.
Ensure: select a channel $chMin$ with the minimum interference.

```
1: chMin = INVALID_CHANNEL;
2: Infer(chMin)=INFINITY;
3: for ch=1; ch < j + 1; ch++ do
4:     Infer(ch)=0;
5:     Ph=0;
6:     for i ∈ N(ch) do
7:         Ph=Ph+P(ch);
8:     end for
9:     Infer(ch)=Infer(ch)+ (1/d^α) × Ph;
10: end for
11: if Infer(ch) == 0 then
12:     chMin=ch;
13:     return chMin;
14: end if
15: if Infer(ch) < Infer(chMin) then
16:     chMin = ch;
17:     Infer(chMin) = Infer(ch);
18: end if
19: return chMin;
```

Distribution of interface information Nodes can advertise their selected channel list with their multi-hop neighbors by exchanging the interface information, similar to the MPR mechanism in OLSR [9]. An example of neighbor information exchange is shown in Fig. 3. By exchanging the interface information between directly connected neighbors, node A first collets the interface configuration information of one-hop neighbors B and D. All nodes will also forward the interface information received to their neighbors. Thus, node A can also obtain the interface information of its two-hop neighbor C and three-hop neighbor E.

4.1.2 Task allocation

Assuming node A with M interfaces in the receiving mode has N neighbors having packets to send to node A. Each neighbor should select one interface to send packets and ideally the N neighbors should be associated evenly to M interfaces, i.e., the load assigned on every interface is balanced.

First, the relative load L_i between A and its neighbor i will be measured by counting the number of the packets in Eq. 3.

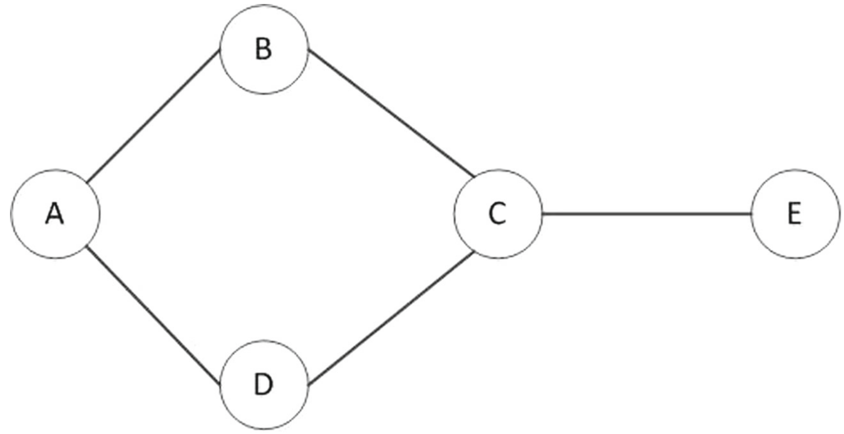
$$L_i = p_i / P, \tag{3}$$

where p_i is the number of packets sent to A by neighbor i and P is the total number of packets that A received from all the neighbors with the window of time measured. The link load is estimated based on the monitoring of the wireless channel status, and we can use an exponentially weighted moving average (EWMA) method to obtain the moving average of the link load. In our design, the packets are with same size. More generally, the packets may have different sizes which cause differenct airtime during the transferring. In that situation, the load for the receiving interface can be represented as bits of received.

Second, we define a subtask as a data stream from the tagged node, and merge the subtasks selecting the same receiving interface of the receiver into one task. N subtasks whose loads are L_1, L_2, ..., L_N respectively will be merged into M tasks. The goal of the merge algorithm is to ensure that M tasks have the minimum variance of all possible combinations. For this, we adopt a merge algorithm based on the Huffman tree as shown in Algorithm 1. and have proved its correctness.

The following is the proof that Algorithm 2 will have the minimum variance. In the first merge, the lowest two loads are L_a and L_b. Without loss of generality, let L_c and L_d represent the loads of another two-task pair. We need to prove that the variance of merging L_a and L_b is smaller than that of merging L_c and L_d. Assume that $\overline{L}$ is the average load for all loads after emerging, and DL is the sum of variance for other tasks' load excepting L_a, L_b, L_c and L_d.

Fig. 3 The example of neighbor information exchange



If merging L_a and L_b, the variance becomes

$$D_{ab} = ((L_a + L_b) - \overline{L})^2 + (L_c - \overline{L})^2 + (L_d - \overline{L})^2 + DL.$$

If merging L_c and L_d, the variance becomes

$$D_{cd} = ((L_c + L_d) - \overline{L})^2 + (L_a - \overline{L})^2 + (L_b - \overline{L})^2 + DL.$$

Compare D_{ab} and D_{cd},

$$Diff = D_{ab} - D_{cd} = 2(L_a L_b - L_c L_d).$$

Since L_a and L_b are the lowest two loads, i.e., $L_a \le L_c$ and $L_b \le L_d$, we have $L_a L_b \le L_b L_c$ and $L_b L_c \le L_c L_d$. Combining them, $L_a L_b \le L_c L_d$ is obtained. Thus, $Diff \le 0$ and $D_{ab} \le D_{cd}$. The minimum variance can be ensured in the first merge. Similarly, in the following merges, the property will be kept. Finally, the merged M tasks with the minimum variance are obtained.

Algorithm 2 Task allocation

Require: N subtasks T_1, $T_2, ..., T_N$ with the load L_1, $L_2, ..., L_N$.
Ensure: M tasks with the minimum variance.
1: **while** $N > M$ **do**
2: select two subtask T_i and T_j with the lowest two loads L_i, L_j;
3: construct a new task T_{i+j} with the load $L_{i+j} = L_i + L_j$;
4: delete the two subtasks T_i and T_j.
5: $N - -$;
6: **end while**
7: return;

4.2 Sending interfaces

The packets to send are arranged in different queues associated with different channels. Every queue will be mapped with a sending interface, which takes turns to send packets in different queues. Ideally, each interface should handle the same number of queues. But other factors also have influence on the performance.

Moreover, a proper value should be set for the queue-switching interval, which is the maximum time that an interface can work continuously on a certain channel. A lengthy interval will increase the delay of other queues while a short one will decrease the interface utilization especially when the switching time for an interface is large. Also, the loads of one-hop neighbors vary all the time. For the neighbor with high-volume traffic, the longer queue-switching interval will be preferable because the interface can be utilized efficiently and the delay is decreased.

To make a tradeoff between the end-to-end delay and the interface utilization, we adopt a method that different interfaces are assigned with different switching interval and the interfaces with a longer interval should be mapped to the busier queues, and the details are given below.

Switching interval of interfaces The sending interfaces are numbered, $1, 2, ..., I_N$, and T_k and N_k represent the switching interval of interface k and the number of mapped queues of interface k respectively. Number 1 interface is a dynamic interface with the switching interval of T, the basic unit. We set the switching interval of interfaces k according to Eq. 4.

$$T_k = T \times 2^{(k-1)}. \quad (4)$$

We then map busy queues to large numbered interfaces with larger queue-switching intervals, to prevent frequent channel switching.

Equation 5 give the queue polling latency D_k of interface k, the maximum recurrence time for the interface returns to a queue.

$$D_k = N_k \times (T_k + delay_s), \tag{5}$$

where $delay_s$ is the channel switching delay of interface k. When there is a new queue to be mapped, the interface with the minimum sending delay will be selected. By adjusting the number of mapped queues, the sending delay of all interfaces can be balanced.

Load level of queues To measure how busy a queue is, we set LV to represent the load level for a queue. The larger the LV_q is, the busier the queue q is.

LV_q is adjusted when the interface leaves the queue. If the switch occurs due to an empty queue, the load is light so that LV_q is decreased by 1; otherwise, LV_q is increased by 1. LV is initialized according to the first mapped interface, as explained below.

Range of the load level of interfaces Every interface sets a range R of load level. Only if the LV_q belongs to $R(k)$, queue q can be mapped with interface k, otherwise it will be adjusted. The range of interface k is set by Eq. 6, where s is a constant to extend the range in order to avoid too frequent adjustment for the queues.

$$R(k) = [s \times (2k-1), s \times (2k+1)). \tag{6}$$

The median of $R(k)$ is $(s \times 2k)$, which is used to initialize the load level of the queues mapped to the interface. In Eq. 6, $s \geq 1$. $s = 1$ means that a node should switch channel once the change of LV_q is greater than 2; with $s = 2$, channel switching is trigged while the change of LV_q is greater than 4, and so on. s should be set with a suitable value to control channel switch frequency.

Since the traffic loads vary dynamically, based on the load level, the queue mapping will be adjusted using Algorithm 3, which ensures that a busy queue is mapped with a long interval interface (Line 7 and 15). The adjustment for the queues is to exchange queues between two consecutively numbered interfaces to ensure that the queue polling delay reaches equilibrium for all interfaces. Further, if a queue needs to be moved to an interface with a longer interval, but the interface does not have a queue needing a shorter interval, the exchange will not happen (Line 9), and vice versa (Line 17).

Algorithm 3 Adjustment operation

Require: interface k switches from queue q; LV_q: the load level of queue q; $R(k)$: the load level range of interface k, $[s \times (2k-1), s \times (2k+1))$
Ensure: adjust the mapped interface with queue q.
1: interface k switches from queue q;
2: **if** $isEmpty(q)$ **then**
3: $LV_q --$;
4: **else**
5: $LV_q ++$;
6: **end if**
7: **if** $LV_q \geq s \times (2k+1)$ **then**
8: select the queue q_{min} with the minimum load level from interface $(k+1)$;
9: **if** $LV_{q_{min}} < s \times 2(k+1)$ **then**
10: map q with interface $k+1$;
11: $LV_q = s \times 2(k+1)$;
12: map q_{min} with interface k;
13: $LV_{q_{min}} = s \times 2k$;
14: **end if**
15: **else if** $LV_q < s \times (2k-1)$ **then**
16: select the queue q_{max} with the maximum load level from interface $(k-1)$;
17: **if** $LV_{q_{max}} > s \times 2(k+1)$ **then**
18: map q with interface $k+1$;
19: $LV_q = s \times 2(k-1)$;
20: map q_{max} with interface k;
21: $LV_{q_{max}} = s \times 2k$;
22: **end if**
23: **else**
24: return;
25: **end if**
26: return;

4.3 Adaptive interfaces

The adaptive interfaces can be used as a sending or receiving interface to balance the load among all interfaces. For example, a data source node may have more packets to send, so more adaptive interfaces will work as sending interfaces; more receiving interfaces are necessary for nodes collecting and aggregating information. Here, the main concern is when and how the adjustments are conducted.

It is obvious that when the sending and receiving loads differ widely, an adjustment is necessary. By monitoring the wireless activities, we estimate the load of the receiving and sending interfaces by Eqs. 7 and 8 respectively.

$$P_{receive} = t_{active} / (T \times N_{receive}), \tag{7}$$

$$P_{send} = 1 - t_{idle}/(T \times N_{send}), \tag{8}$$

where $N_{receive}$ and N_{send} are the number of the receiving and sending interfaces respectively, t_{active} is the total active of all receiving interfaces and t_{idle} is the idle time of all sending interfaces. Because the switching delay is hard to monitor so the idle time is adopted. The difference of the loads is $D_P = P_{receive} - P_{send}$. Th is a threshold used to avoid too frequent adjustments. If $|D_P| > Th$, an adjustment is deemed necessary.

If $D_P > Th$, meaning higher receiving load, one interface should change from the sending mode to the receiving mode; and if $D_P < Th$, one more sending interface is needed.

The two processes need some common steps but with different orders. For adding a receiving interface, the steps are shown as follows: (a) identify the interface that will change; (b) re-adjust the maps between the queues and the sending interfaces; (c) re-assign the tasks by Algorithm 2; (d) inform the neighbors of the receiving channel and task allocations; For the new receiving interface, one more step is necessary: (e) select a working channel for the new interface by Algorithm 1.

For changing from sending to receiving, the steps are (a) (b) (c) (d). In (a), the highest interface will be better for it maps with the least queues and less queues need to be re-mapped, so a very lengthy switching interval can be avoided. In (b), the queues mapped with the changed interface is mapped to other interfaces in turn.

For changing from receiving to sending, steps (a) (c) (b) (d) are conducted successively. In (a), the interface with the least load will be selected because it is least affected for task allocation. In (b), the new interface is set with the highest number and move the queues from the interface with the highest sending delay to it until it is no longer with the least sending delay, so the sending delay can be similar for every interface.

4.4 Cooperation among different type of interfaces

The proposed protocol is deployed in all the nodes and controls the data transferring with distributed mode. For multi-hop wireless mesh networks, the number of sink nodes is often limited and much less than the mesh routers or relays, while most traffic converges toward the sink nodes who are connected to the Internet with wired links. Uplink and downlink data are received and sent by wireless interfaces of the sink nodes with high-load level compared with other mesh routers. The protocol can only balance the load of receiving and sending traffic among the interfaces, yet it cannot balance the traffic between nodes. Consequently, it is not helpful to the global optimization, especially, the sink nodes and those nodes close to sinks. We need the upper-layer routing protocol to decide better paths for load balancing, which will be further investigated in the following performance evaluation section.

5 Performance evaluation

We evaluated the performance of the proposed link protocol, LBLP, by NS3-19 [23] in Linux. The MAC protocol adopted is IEEE 802.11a, which needs no modification except the added module for statistics of the MAC status.

5.1 Simulation setting

The IEEE 802.11 a standard specifies 12 non-overlapping channels, which have no mutual interference in theory. However, due to technology limitation, they may not always be interference-free to each other. Authors in [21] showed that the interfaces working on adjacent non-overlapping channels may also interfere with each other. Kyasanur and Vaidya [3] found that there are only 5 or 6 orthogonal channels available. With the advance of the technologies, more orthogonal channels may be possible. Thus, we set the number of orthogonal channels to vary from 5 to 12.

The interface switching delay refers to the interval for an interface switching from one channel to another, during which the interface cannot send or receive any packet. In theory, the delay could be as low as tens of microseconds [4]; currently, the switching delay of the IEEE 802.11 hardware is estimated to be about several milliseconds [22]. Hence, the switching delay in our experiments is 2 ms.

As for the number of the interfaces equipped with a node, it can differ from each other. Given the multi-interface framework considered in Fig. 1, in our simulation, the interface number ranges from 2 to 5, which is smaller than the available orthogonal channels. For all interfaces, the transmission range is 250 m and the carrier sense range is 500 m. Other parameters are shown in Table 1. The routing protocol is the well-known AODV for its simplicity. It can work with LBLP without any modification. The results presented here are the average of 10 simulation runs with different random seeds.

Next, we mainly investigate the impacts of the number of channels and interfaces, and compare the network throughput performance of the proposed LBLP with existing protocols.

5.2 Number of the channels and interfaces

The performance of LBLP is largely dependent on the number of channels and interfaces as they directly affect the interference. Set a 10×10 mesh network with the distance of 200m between the adjacent nodes. A pair of the source node

Table 1 Simulation paraments setting

Parameter	Value
MAC standards	IEEE 802.11a
Sending power	18 dBm
Switching delay	2 ms
Transmission range	250 m
Carrier sense range	500 m
Propagation delay model	Constant speed propagation delay mode
Propagation loss model	Log distance propagation loss mode
RTS/CTS	Enable
Unit switching interval	5 ms
Orthogonal channels	5–12
Interfaces number	2–5
Simulation time	200 s
Simulation runs	10

and the destination node will transfer FTP traffic over TCP, and 400 pairs are selected randomly. We set the channel number vary from 5 to 12 and the interface number vary from 2 to 5, and the network throughput is shown in Fig. 4.

The results in Fig. 4 show that with the increasing number of channels, the throughput with more interfaces increases but the growth is slowing down. The reason is that the link layer protocol can utilize well the channel resources. But with the limit of the number of interfaces, the number of channels that can be utilized is limited so the growth slows down. In addition, more channels may cause more channel switchings which lead to a higher overhead. As for the number of interfaces, we can see that more interfaces result in a higher throughput with the same number of channels. More interfaces and channels can both improve the network throughput but they are limited by each other, i.e., more channels can not always bring an increased throughput without enough number of interfaces, and vice versa.

5.3 Protocol comparison

We study and compare the protocol performance using three settings: (a) LBLP with AODV [14], called LAR, (b) LBLP with WCETT [21], called LWR, (c) AODV-MR [15], a cross-layer solution combining the link and routing protocols.

In the simulations, 100 nodes are randomly distributed in a square area of $1000 \times 1000\ m^2$. All the nodes are equipped with 4 interfaces, and 12 orthogonal channels are available. The traffic is FTP over TCP. Two groups of simulations were conducted.

In the first scenario, we consider the source/destination pairs uniformly selected among all nodes, and change the number of pairs from 50 to 500 to show the performance of low traffic load and high traffic load settings. The results are shown in Fig. 5. From the figure, performance of LBLP with AODV and WCETT are very close, and AODV-MR performs worse. The routing protocol WCETT does not consider the distribution of the channels and the overhead of the channel switching, so it does not perform much better than LAR (LBLP with AODV). Besides, with the increase of the number of concurrent flows, most of the channels and interfaces will be utilized and their throughputs tend to be similar. AODV-MR performs worse due to the lack of load balance in its simple channel assignment.

In the second scenario, five nodes are set as the gateways which have much higher traffic loads than the others. This is a realistic scenario considering the non-uniform traffic patterns. The other nodes will randomly select a nearby gateway to access the Internet, and all the traffic will converge to the gateway. The average throughput is presented in Fig. 6.

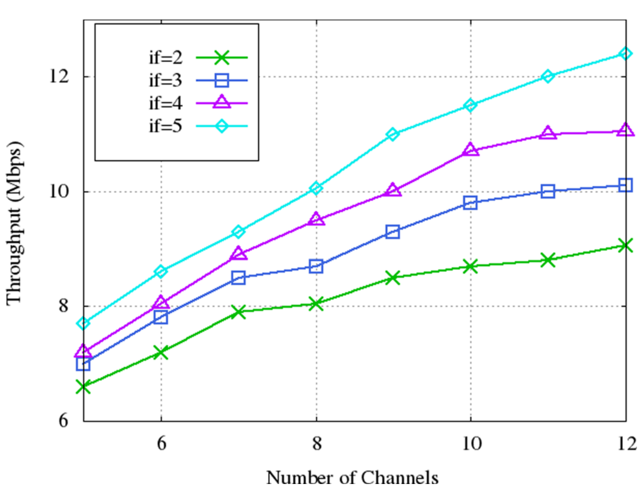


Fig. 4 Throughputs with different number of channels and interfaces

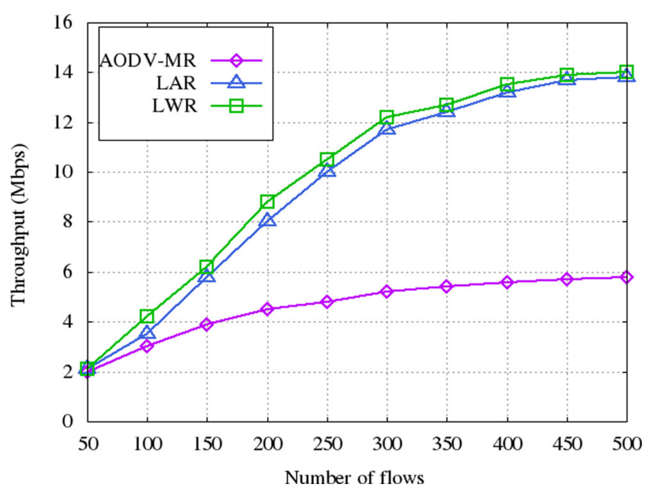


Fig. 5 Throughput comparison, random source/destination-pair case

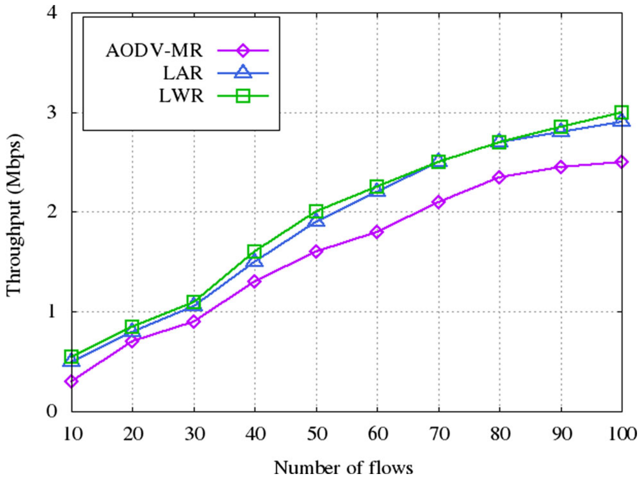


Fig. 6 Throughput comparison, five-gateway case

The results are similar to that of the first group. LAR and LWR have the similar performance, and they outperform AODV-MR. When the number of concurrent flows is small, the performance of LAR or LWR is similar to AODV-MR. While with the more flows, the load balance of LBLP improve the overall performance more substantially.

The simulation results of both scenarios show that the proposed LBLP can balance the traffic load according to the dynamic traffic with uniform or non-uniform traffic patterns to achieve a higher throughput, and it can work well with the existing routing protocols.

6 Conclusion

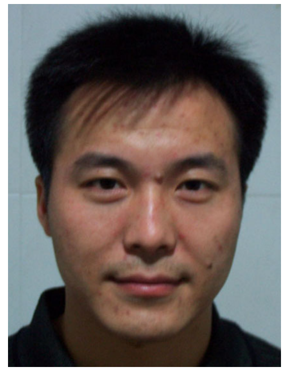
In this paper, we have proposed a link protocol, LBLP, for multi-channel and multi-interface wireless mesh networks. The protocol focuses on the channel assignment and the link scheduling, aiming to maximize the network throughput by balancing the traffic load. It further considers how to mitigate the overheads due to large switching delay. Simulation results have shown that LBLP can out-perform the existing solutions. There are still many open issues beckoning further investigation. In this work, how to quantify the performance given by the number of channels and interfaces, and how to optimize the adjustments of interfaces remains open. Some other routing protocols, such as ETT, WCETT, iAware, may need to be modified to work with the link protocol, which requires further research.

Acknowledgements The research is supported in part by grants from the National Natural Science Foundation of China (61073186, 61272494 and 61379058), grant No. 2017zzts482 from Central South University and grants from Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Raniwala A, Gopalan K, Chiueh T-c (2004) Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. ACM SIGMOBILE Mob Comput Commun Rev 8(2):50–65
2. Ramachandran KN, Belding EM, Almeroth KC, Buddhikot MM (2006) Interference-aware channel assignment in multi-radio wireless mesh networks. IEEE INFOCOM 6:1–12
3. Kyasanur P, Vaidya NH (2006) Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. ACM SIGMOBILE Mob Comput Commun Rev 10(1):31–43
4. Bahl P, Chandra R, Dunagan J (2004) SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In: Proceedings of the 10th annual international conference on Mobile computing and networking. ACM, pp 216–230
5. Raniwala A, Chiueh T-c (2005) Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings IEEE, vol 3. IEEE, pp 2223–2234
6. Dong M, Ota K, Liu A, Guo M et al (2016) Joint optimization of lifetime and transport delay under reliability constraint wireless sensor network. IEEE Trans Wirel Commun 27(1):225–236
7. Ren J, Zhang Y, Deng R, Zhang N et al Joint channel access and sampling rate control in energy harvesting cognitive radio sensor networks. In: IEEE Transactions on Emerging Topics in Computing, to appear. https://doi.org/10.1109/TETC.2016.2555806
8. Ren J, Zhang Y, Liu A, Zhang N et al (2016) Dynamic channel access to improve energy efficiency in cognitive radio sensor networks. IEEE Trans Wirel Commun 15(5):3143–3156
9. Clausen T, Jacquet P, Adjih C (2003) Optimized link state routing protocol (OLSR)
10. Nasipuri A, Zhuang J, Das SR (1999) A multichannel csma mac protocol for multihop wireless networks. In: Wireless Communications and Networking Conference, 1999. WCNC IEEE. IEEE, p 1999
11. Alicherry M, Bhatia R, Li LE (2005) Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In: Proceedings of the 11th annual international conference on Mobile computing and networking. ACM, pp 58–72
12. Xiao L, Lin WS, Chen Y et al (2012) Indirect reciprocity security game for large-scale wireless networks. IEEE Trans Inf Forensic Secur 7(4):1368–1380
13. Si W, Selvakennedy S, Zomaya AY (2010) An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks. J Parallel Distrib Comput 70(5):505–524
14. Perkins CE, Royer EM (1999) Ad-hoc on-demand distance vector routing. In: Second IEEE Workshop on Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. IEEE, pp 90–100
15. Pirzada AA, Portmann M, Indulska J (2008) Performance analysis of multi-radio aodv in hybrid wireless mesh networks. Comput Commun 31(5):885–895
16. Gong MX, Midkiff SF (2005) Distributed channel assignment protocols: a cross-layer approach [wireless ad hoc networks]. In: Wireless Communications and Networking Conference, 2005 IEEE, vol 4. IEEE, pp 2195–2200

17. Subramanian AP, Buddhikot MM, Miller S (2006) Interference aware routing in multi-radio wireless mesh networks. In: 2nd IEEE Workshop on Wireless Mesh Networks WiMesh 2006. IEEE, p 2006
18. Jiang W, Liu S, Zhu Y, Zhang Z (2007) Optimizing routing metrics for large-scale multi-radio mesh networks. In: WiCom International Conference on Wireless Communications, Networking and Mobile Computing, 2007. IEEE, p 2007
19. Yang Y, Wang J, Kravets R (2005) Designing routing metrics for mesh networks. In: IEEE Workshop on Wireless Mesh Networks (WiMesh)
20. Mogaibel HA, Othman M, Subramaniam S, Asilah Wati Abdul Hamid N (2012) On-demand channel reservation scheme for common traffic in wireless mesh networks. J Netw Comput Appl 35(4):1329–1351
21. Draves R, Padhye J, Zill B (2004) Routing in multi-radio, multi-hop wireless mesh networks. In: Proceedings of the 10th annual international conference on Mobile computing and networking. ACM, pp 114–128
22. Chandra R, Bahl P (2004) Multinet: Connecting to multiple ieee 802.11 networks using a single wireless card. In: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, vol 2. IEEE, pp 882–893
23. Network Simulator NS-3. http://www.nsnam.org/
24. Deng X, Liu Q, Li X, Cai L, Chen Z (2013) A Load Balance Link Layer Protocol for Multi-channel Multi-interface Wireless Mesh Networks. In: *High* Performance Computing and Communications IEEE. IEEE, p 2013
25. Har D, Xia HH, Bertoni HL (1999) Path-loss prediction model for micro-cells. IEEE Trans Veh Technol 48(5):1453–1462
26. Xiao L, Liu J, Li Q et al (2015) User-centric view of jamming games in congtive radio networks. IEEE Trans Inf Forensic Secur 10(12):2578–2590
27. Xie K, Wang X, Liu X et al (2016) Cooperative routing with relay assignment in multi-radio multi-hop wireless networks. IEEE Trans Comput 24(2):859–872
28. Xie K, Wang X, Liu X et al (2016) Interference-aware cooperative communication in multi-radio multi-channel wireless networks. IEEE Trans Comput 65(5):1528–1542
29. Gui J, Zhou K (2016) Flexible adjustments between energy and capacity for topology control in heterogeneous wireless multi-hop networks. J Netw Syst Manag 24(4):789–812
30. Gui J, Zhou K, Xiong N (2016) A cluster-based dual-adaptive topology control approach in wireless sensor networks. Sensors 16(10):1576
31. Zhang G, Yang K, Jiang H et al (2016) Equilibrium price and dynamic virtual resource allocation for wireless network virtualization. In: Mobile Networks and Applications. https://doi.org/10.1007/s11036-016-0766-9
32. wang K, Yang K, Chen H et al (2017) Computation diversity in emerging networking paradigms. IEEE Wirel Commun 24(1): 88–94
33. Zhang D, Chen Z, Awad MK, Zhou H, Zhang N, Shen X (2016) Utility-optimal resource management and allocation algotithm for energy harvesting congtive radio sensor networks. In: IEEE Journal on Selected Areas in Communications, vol 34. IEEE, pp 3552–3565
34. Zhang D, Chen Z, Ren J, Zhang N, Awad MK, Zhou H, Shen X (2017) Energy-harvesting-aided spectrum sensing and data transmission in heterogeneous congtive radio sensor network. In: IEEE Transactions on Vehicular Technology. IEEE, pp 831–843



Xiaoheng Deng received the Ph.D. degrees in computer science from Central South University, Changsha, Hunan, P.R. China, in 2005. Since 2006, he has been an Associate Professor and then a Full Professor with the Department of Electrical and Communication Engineering, Central South University. He is the Chair of RS Changsha Chapter, a senior member of CCF, a member of CCF Pervasive Computing Council, a member of IEEE and ACM. He has been a chair of CCF YOCSEF CHANGSHA from 2009 to 2010. His research interests include wireless communications and networking, congestion control for wired/wireless network, cross layer route design for wireless mesh network and ad hoc network, online social network analysis.

Jie Luo is a M.Sc. student in school of Information Science and Engineering of Central South University, Changsha, China. He received the B.Sc. degrees in electronic and information from Southwest University, Chongqing, China, in 2016. His major research interests are wireless network and edge computing.

Lifang He received her M.Sc. degrees in Information Science and Engineering from Central South University, Changsha, China, in 2016, B.Sc. degrees in electronic and information engineering from Kunming University of Science and Technology, Kunming, China, in 2013. Her major research interests are wireless mesh network and smart grid.

Qiang Liu CCF member, received the M.Sc. degrees in Information Science and Engineering of Central South University, Changsha, China, in 2014. His research interests include the routing protocol design and channel assignment of the wireless mesh network.

Xu Li CCF member, received his M.Sc. degrees in Information Science and Engineering of Central South University, Changsha, China, in 2014. His research interests include cross-layer design in routing metric and protocol of wireless mesh network and MAC layer.

Lin Cai received her M.A.Sc. and PhD degrees (awarded Outstanding Achievement in Graduate Studies) in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2002 and 2005, respectively. Since 2005, she has been with the Department of Electrical & Computer Engineering at the University of Victoria, and she is currently a Professor. Her research interests span several areas in wireless communications and networking, with a focus on network protocol and architecture design supporting emerging multimedia traffic over wireless, mobile, ad hoc, and sensor networks. She has been a recipient of the NSERC Discovery Accelerator Supplement Grant in 2010 and in 2015, and the best paper awards of IEEE ICC 2008 and IEEE WCNC 2011. She has served as a TPC symposium co-chair for IEEE Globecom'10 and Globecom'13, and an Associate Editor for IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, EURASIP Journal on Wireless Communications and Networking, International Journal of Sensor Networks, and Journal of Communications and Networks (JCN).