Distributed Joint Channel Assignment, Routing and Scheduling for Wireless Mesh Networks ⋆

Xin Wang [a,*] J.J. Garcia-Luna-Aceves [a,b]

[a] *Department of Computer Engineering, University of California, Santa Cruz*

1156 high street, Santa Cruz, CA 95064, U.S.A.

Phone: 1-831-4595436, Fax: 1-831-4594829

[b] *Palo Alto Research Center (PARC)*

3333 Coyote Hill Road, Palo Alto, CA 94304, U.S.A.

Abstract

We present the JARS (Joint channel Assignment, Routing and Scheduling) scheme for ad hoc wireless networks in which nodes are endowed with multiple radios. JARS is one example of the benefits gained by the integration of routing, scheduling, and channel assignment by using the multiple radios at each node to transmit and receive simultaneously on different orthogonal channels. Instead of choosing the optimal route based on the predetermined transmission scheduling and channel assignment results, JARS incorporates the efficiency of underlying channel assignment and scheduling information into the routing metric calculation so that the route with the maximal joint spatial and frequency reuse is selected. Once a path is established, the channel assignment and link scheduling are also determined at the same time. JARS also adapts different channel assignment and scheduling strategies according to the different communication patterns of broadcast and unicast transmissions. Simulation results show that JARS efficiently exploits the channel diversity and spatial reuse features of a multi-channel multi-radio system.

Key words: Routing, scheduling, channel assignment, multi-channel multi-radio, logical distance, transmission fraction

1 Introduction

The multi-channel multi-radio system is introduced to exploit the frequency diversity of the wireless networks. By assigning the interference links with orthogonal channels, multi-channel multi-radio system can dramatically reduce the wireless interference that widely exists in the classical multi-hop wireless networks and enhance the network capacity. In multi-channel multi-radio networks, channel assignment, routing and scheduling are dependent on each other. The input of any component is partially decided by the output of the other two components, e.g. 1).**transmission scheduling**: channel assignment decides whether two links are interfere with each other, and route selection decides which links will be used for transmissions; 2).**channel assignment**: different transmission scheduling will generate different interference link set, it further decides which links need to be assigned with orthogonal channels; good route selection balance the traffic throughout the network so that the

⋆ This work was supported in part by the Baskin Chair of Computer Engineering at UCSC, the National Science Foundation under Grant CNS-0435522, and the U.S. Army Research Office under grant No.W911NF-041-1-0224.
∗ Corresponding author
Email addresses: wangxin@soe.ucsc.edu (Xin Wang), jj@soe.ucsc.edu (J.J. Garcia-Luna-Aceves).

channel diversity can be utilized to the largest extent; 3).**route selection**: channel assignment decides the network topology, which influences the route selection results directly; since routing control packets are transmitted as the data packets at the MAC layer, the transmission scheduling decides how the routing information is propagated throughout the network.

However, the correlations between these three components are at different timescales. We note that channel assignment and scheduling are formed based on the two-hop information, while route selection are made based on the end-to-end information between the traffic source and destination. So channel assignment and scheduling are coupled with each other more tightly at small timescales (a few packet transmissions), while route selection interacts with the other two components at large timescales (hundreds of packet transmissions).

Based on the above discussions, in order to fully leverage the spatial and frequency diversity of multi-channel multi-radio networks, we need to jointly consider the channel assignment, routing and scheduling problem. There are several challenges in designing a fully distributed joint channel assignment, routing and scheduling algorithm. First, how to evaluate a transmission schedule is efficient in terms of both channel diversity and spatial reuse. Second, how MAC layer and network layer interact correctly to exploit the frequency diversity and spatial reuse at both layer.

In this paper, we propose the distributed **J**oint channel **A**ssignment, **R**outing and **S**cheduling protocol for wireless mesh networks (JARS). Due to different characteristics of broadcast and unicast transmissions, JARS adapts different channel assignment and link scheduling strategies according to packet types.

For broadcast transmissions, JARS requires all nodes in the communication range to converge on a common channel at a specific time slot, which allows the broadcast transmissions propagate throughout the network through the efficient utilization of the broadcast nature of wireless medium. For unicast transmissions, we introduce a unified metric, *transmission fraction*, to evaluate the efficiency of the joint channel assignment and link scheduling in terms of spatial and frequency reuse. The transmission fraction is used to replace the traditional link cost in the route distance calculation. Through this approach, the route which most efficiently utilizes the spatial and frequency reuse is selected.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 explains the motivation for JARS. Section 4 introduces the details of the proposed approach. Section 5 numerically analyzes the approximate MAC layer throughput and complexity of routing protocol. Section 6 evaluates the performance of JARS through simulations. Section 8 concludes the paper.

2 Related Work

Gupta et al. [1] study the asymptotic capacity of single channel, single radio multi-hop wireless network under two interference models, *protocol model* and *interference model*. Kyasanur et al. [2] further investigate the impact of number of channels and radios on the asymptotic capacity. They show that the capacity of multi-channel networks exhibits different bounds that are dependent on the ratio between number of channels and number of radios. It may be possible to build capacity-optimal multi-channel networks with as few as one interface per node.

Joint routing, scheduling and channel assignment [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] has been proposed to utilize the channel diversity of the multi-channel multi-radio networks. Raniwala et al. [3] propose a centralized channel assignment and routing algorithm, which uses an heuristic approach to obtain a static channel assignment. An improved distributed channel assignment algorithm is proposed in [4]. Kyasanur et al. [5] propose an interface assignment strategy where the number of available interfaces is less than the number of available channels. It fixes a channel on one radio and switches channels on other radios. Nodes can communicate with each other through the fixed common radio without requiring specialized coordination algorithms.

Kodialam et al. [6] consider the problem of jointly routing and scheduling transmissions to achieve a given rate vector. They use a simple interference model, which is derived from the CDMA based multi-hop networks to map the scheduling problem to edge coloring problem. They have proven that their solution is within $\frac{2}{3}$ of the optimal solution. Zhang et al. [7] formalize the problem for joint routing and channel switching in wireless mesh networks and use column generation method to solve the problem. Alicherry et al. [8] mathematically formulate the joint channel assignment and routing problem for infrastructure wireless mesh networks. They aim to maximize the bandwidth allocated to each traffic aggregation point subjected to fairness constraint and propose a constant approximation algorithm for this NP-hard problem. Kodialam et al. [9] develop a network model that characterizes the channel, radio and interference constraint in a fixed broadband wireless network, which provides necessary and sufficient conditions for a feasible channel assignment and schedule. Meng et al. [10] formulate the joint routing and channel assignment problem based on radio and radio-to-radio link. They introduce a *scheduling graph* and derive a sufficient condition for the feasibility problem of time fraction. Tam et al. [11] propose a joint multi-channel and multi-path control protocol (JMM). JMM coordinates channel usage among slots using a receiver-based channel assignment and schedules transmissions along dual paths. JMM uses a routing metric which explicitly accounts for the disjointness between paths and interference among links to select two maximally disjoint paths. Wu et al. [12] propose a channel cost metric (CCM) which reflects the interference cost and channel diversities. Based on CCM, a distributed joint channel assignment and routing protocol is proposed.

JARS is distinct from the previous works in the following aspects:

(1) JARS utilizes the broadcast characteristic of wireless channel and the channel diversity of the multi-channel multi-radio system to achieve the efficient broadcast and unicast transmissions, respectively

(2) Previous works about the distributed algorithm for joint routing, scheduling and channel assignment [11] [12] mainly focus on the selection of the *channel diverse routing*, which is based on the underlying predetermined channel assignment and scheduling results. The routing metrics are mainly derived from WCETT [13] [14], which is a measurement-based routing metric and may be inaccurate for self-traffic [15]. JARS incorporates the channel assignment and transmission scheduling into the *logical distance* of routing metric, the optimal path is chosen based on all possible channel assignment and transmission schedule combinations.

3 Motivation for JARS

In this section, we first explain why we use multiple half-duplex radios to emulate a full-duplex system. Then we introduce the channel division between the control plane and data plane.

3.1 Full-duplex radio

The constraint that nodes can just transmit/receive at one half-duplex radio each time is a primary assumption for the protocol design of wireless networks. However, this constraint reduces the efficient utilization of channel diversity of multi-channel multi-radio system. Consider the following example shown in Figure 1, there are two traffic flows: $H \rightarrow K$ and $A \rightarrow E$, and there are two paths from A to E, $P_1 : (A \rightarrow G \rightarrow F \rightarrow E)$, $P_2 : (A \rightarrow B \rightarrow C \rightarrow D \rightarrow E)$. P_1 has a shorter path length, but it is in the interference range of P_3.

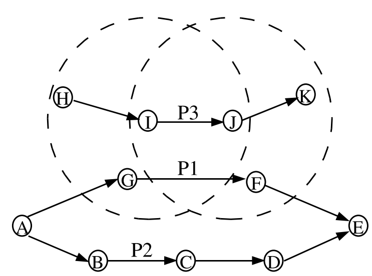


Fig. 1. Schedule comparison between full-duplex and half-duplex radios

If path P_1 and P_3 are selected at the same time, the system can at most allow three links $((HI, GF, JK)$ or $(AG, IJ, FE))$ to transmit simultaneously through assigning orthogonal channels to different links. In other words, at most three channels can be used no matter how many channels are available. Selection of a path with more channel diversity (e.g. P_2) may increase the system throughput, but also implies longer end-to-end delay, which is due to two reasons: 1). more path hops; 2). channel switching overhead, which ranges from 80μs to a few hundred microseconds. For a half-duplex multi-channel multi-radio system, the throughput improvement due to the channel diversity is at the cost of increasing end-to-end delay. Now we assume each node in Figure 1 has two radios which can transmit and receive at the same time, then all links over path P_1 and P_3 can transmit simultaneously through using up to 6 channels. Compared with the half-duplex system, a full-duplex system can utilize the channel diversity more efficiently, and it improves the throughput and end-to-end delay at the same time. On the other hand, previous analysis about the throughput upper bound of the mesh networks using joint routing and scheduling [7] [8] [10] have shown that the system throughput does not increase linearly with the increase of channels, which is due to the fact that now the system bottleneck is that nodes cannot transmit and receive at the same time. If the total system time is dedicated to traffic delivery to or from some bottleneck nodes, performance cannot be further improved by the increase of channels. This also motivates us to introduce a full-duplex system. Based on the above discussions, we use multiple half-duplex radios to emulate a full-duplex node. Full-duplex node operations can be accomplished by assigning the radios of each node to different orthogonal channels and have some of them in receive mode while others are in transmit mode.

3.2 Channel division

We divide the channels into two groups: one control channel and multiple data channels. We note there are several problems for multi-channel MAC protocols without a common control channel: 1). It cannot send broadcast transmissions efficiently. The broadcast packets either need to be sent over all channels, or sent during the rendezvous interval on a common channel (e.g. SSCH [16]). The previous approach incurs high overheads, while the latter approach will increase the transmission delay of control packets; 2). When two nodes are not assigned with any common channels, even if they may in the communication range of each other, transmission failures may be mistaken as link breakage, and can adversely affect the performance of higher layer protocols.

Based on the above discussions, we use a separate control channel to exchange topology information and traffic flow information (the detailed information will be introduced in Section 4). We are aware that the control channel may become the system bottleneck [17]. We address this problem by differentiating the transmissions of control packets (it will be discussed in detail in Section 4.2 and Section 4.3).

4 Distributed joint channel assignment, routing and scheduling

4.1 Assumptions

We assume there are K interfaces that can transmit/receive simultaneously on M orthogonal channels ($K \leq M$). We assume a time frame is made up of multiple time slots. Each node is synchronized on slot systems and nodes access the channel based on slotted time boundaries. Each time slot is numbered relative to a consensus starting point.

4.2 Packet queues, scheduler and radio interface interaction

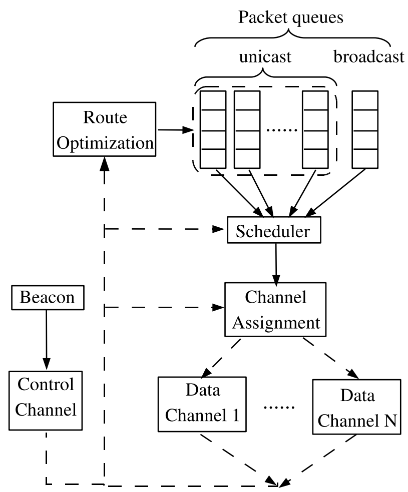


Fig. 2. Packet queues, scheduler and radio interface interaction

Figure 2 illustrates the interaction between packet queues, schedulers and radio interfaces. Each node periodically sends the beacons over the control channel which includes the following items: 1). *neighbor information*: the one-hop neighbor list and two-hop neighbor list of each node i; 2). *traffic flow information*: the existing transmission schedule ($TS(c,t)$) of data channel c at time slot t ($c \in \{c_1, \ldots, c_{M-1}\}, t \in \{1, \ldots, D\}$).

In JARS, there is a specific packet queue for broadcast packets. For unicast transmissions, since neighbors of a node may locate on different channels, packets to each neighbor should be scheduled individually, unicast packets queues are maintained as per-neighbor FIFO queues. Each neighbor is identified with its unique MAC address.

Based on the information collected on the control/data channels, the scheduler selects the packets from the corresponding packet queue, then transmit on data channel c at the reserved time slot t.

4.3 Channel assignment and transmission scheduling strategy

Although all packets received from the network layer are transmitted as the data frames at the MAC layer, due to the different communication patterns of broadcast and unicast transmissions, different channel assignment and transmission scheduling strategies should be adapted. For example, when on-demand routing protocols send route requests to search for paths, we want all nodes in the communication range to receive the broadcast packets. In other words, channel diversity is not beneficial for the local broadcast (we define the broadcast in the one-hop range as *local broadcast*). On the other hand, when we send the unicast packets, we try to maximize the frequency reuse through assigning different links with different channels. Based on the above discussions, the detailed channel assignment and transmission scheduling schemes are as follows:

Broadcast: The source will send a broadcast request packet over the control channel, which reserves time slot t on data channel c. All nodes in the communication range will switch one of its radio interface to channel c at time t to receive the broadcast packet. We utilize the broadcast nature of the wireless medium to achieve the efficient local broadcast. On the other hand, multiple local broadcasts can be sent simultaneously on different channels, which reduces the broadcast collisions.

Unicast: After receiving the $TS(c, t)$, which includes the schedule and channel assignment for broadcast transmissions, nodes will allocate the rest available spectrum resources for unicast transmissions. We introduce a unified metric, *transmission fraction* to evaluate the efficiency of the joint channel assignment and link scheduling. The transmission fraction of link l (TF_l) is defined as the the maximal bandwidth link l can obtain through joint channel assignment and link scheduling. We use TF_l to replace the link cost information used in the traditional routing distance calculation, and get the *logical distance* (LD_p^l) of path p if link l is included. The route that has the minimum logical distance, which means it can achieve the maximum end-to-end throughput is selected. Once a path is selected, the transmission scheduling and channel assignment which are incorporated in TF_l are also established at the same time.

We use a proactive distance-vector routing protocol as the network layer and modified the transmissions of neighbor update messages. During the process of route establishment, the neighbor update messages are transmitted as broadcast packets. After the route is established, the neighbor update messages that include the future scheduling and channel assignment information are transmitted as unicast packets to the specific neighbors (the detailed procedure will be discussed in Section 4.6 and Section 4.7). Now we introduce how we calculate the transmission fraction and logical distance, respectively.

4.4 Transmission fraction calculation

We classify the links in the two-hop range of link (u, v) into two sets according to the link scheduling constraints:

a). Interference links: We denote the interference range of each node as I_R. We define two distinct links $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ are *interfere* if at least one the two pairs $(u_1, v_2), (u_2, v_1)$ are at most I_R apart. In order to transmit simultaneously, two interference links need to be assigned with different channels and there needs to be available radio interfaces at each node. We denote the interference link set of link (u, v) as $I_{(u,v)}$.

b). Two-hop links: This type of links are more than two hops away with each other. We define two distinct links $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ are *two-hop links* if none of the two pairs $(u_1, v_2), (u_2, v_1)$ are in the interference range of each other. Two-hop links can be scheduled at the same time slot with (u, v) to increase the spatial reuse of the system. We denote the two-hop link set of link (u, v) as $T_{(u,v)}$, it is the complementary set of $I_{(u,v)}$.

For each link (u, v), we first generate $I_{(u,v)}$ and $T_{(u,v)}$ based on the topology information collected on the control channel as input, then use a greedy algorithm to obtain the maximal transmission fraction each link could obtain, as Figure 3 shows. We denote $L^2_{(u,v)}$ as the link set in the two-hop range of (u, v) (including (u, v)), and $NT_{(m,n)}$ is the number of scheduled transmissions for link (m, n). At each time slot t, the links with minimum NT is scheduled for transmission to achieve the fairness among different links. Link (u, v) is scheduled with $T_{(u,v)}$ to increase the spatial-reuse of the system. At each step of scheduling, we not only assign the data channels, but also the radio interfaces of the corresponding nodes. Given there are totally $M - 1$ data channels and D slots in each frame, the total number of transmission opportunities are $(M - 1)D$. Each link in the two-hop range of (u, v) can obtain at least $min_{(m,n)\in L^2_{(u,v)}}\{NT_{(m,n)}\}$ transmission opportunities. The transmission fraction of link (u, v) can obtain is $TF_{(u,v)} = \frac{min_{(m,n)\in L^2_{(u,v)}}\{NT_{(m,n)}\}}{(M-1)D}$.

PSEUDO-CODE OF JOINT CHANNEL ASSIGNMENT AND LINK SCHEDULE ALGORITHM

Notations:
$NR_i(t)$: number of available radios interfaces of node i at time slot t;
$CS_i(t)$: available data channel set of node i at time slot t;
NT_l: number of scheduled transmissions for link l;
$TS(c,t)$: transmission schedule for channel c at time slot t, it is initialized by the information collected on the control channel;
$I((m,n),(u,v))$: indicator function, $I((m,n),(u,v)) = 1$ if (m,v) or (u,n) is at most I_R away.

Input:
(u,v), $I(u,v)$, $T(u,v)$;

```
procedure INITIALIZATION
    NR_i(t) ← K;    ▷ for each node i at every time slot t
    CS_i(t) ← {c_1, ..., c_M};
    NT_l ← 0;                         ▷ for each link l
end procedure

procedure CHECK_CHANNEL_RADIO(m, n)
    if (NR_m > 0) ∧ (NR_n > 0) ∧ ((CS_m ∪ CS_n) ≠ ∅) then
    /*There are available radio interfaces and common data channel*/
        for each c_j ∈ (CS_m ∪ CS_n) do
            if ∀l ∈ TS(c_i, t), I(l, (m,n)) == 0 then
                return c_j;
            end if
        end for
    end if
    return ∅;
end procedure

procedure UPDATE(NS, LS, c_j)
/*NS: Selected node set;*/
/*LS: Selected link set;*/
    for each node i ∈ NS do
        NR_i(t) = NR_i(t) − 1;
        CS_i(t) = CS_i(t) − {c_j};
    end for
    for each link j ∈ LS do
        NT_j = NT_j + 1;
    end for
end procedure

procedure CHANNEL ASSIGNMENT AND LINK SCHEDULING
    for each time slot t (t ∈ 1, ..., D) do
        for each link (m,n) ∈ {(u,v) ∪ T_(u,v)} do
▷ /*Schedule link (u,v) and its two-hop links*/;
            if NT_(m,n) = min_{l ∈ L²_(u,v)} {NT_l} then
                if (c_j = check_channel_radio(m,n)) ≠ ∅ then
                    TS(c,t) = TS(c,t) ∪ (m,n);
                    update(m ∪ n, (m,n), c_j);
                end if
            end if
        end for
        for each link (m,n) ∈ I_(u,v) do        ▷
/*Schedule the interference links*/;
            if NT_(m,n) = min_{l ∈ L²_(u,v)} {NT_l} then
                if (c_j = check_channel_radio(m,n)) ≠ ∅ then
                    TS(c,t) = TS(c,t) ∪ (m,n);
                    update(m ∪ n, (m,n), c_j);
                end if
            end if
        end for
    end for
end procedure
```

Fig. 3. The specification of transmission fraction calculation algorithm

4.5 Logical distance calculation

The *logical distance (LD)* of path p is given by a path function f^p based on the transmission fraction of its consisting links. We define the f^p as $f^p = min_l(\frac{1}{TF_l}), l \in p$.

Let LD^i_j denote the logical distance from node i to destination j as known by node i. LD^i_{jk} denotes the logical distance LD^k_j from node k, which is a neighbor of node i, to destination j, as reported to node i by node k. FLD^i_j denotes the *feasible logical distance (FLD)* of node i for destination j, which is an estimate of the minimal logical distance maintained for destination j by node i.

A node i that runs JARS maintains a routing entry for each destination j, which includes FLD^i_j, LD^i_j and the successor set chosen for j (denoted by S^i_j) . Node i maintains a neighbor table that records the logical distance LD^i_{jk} reported by each node k in its neighbor set N^i for each destination j; and a link table that reflects the transmission fraction $TF_{(i,k)}$ for each adjacent link $(i,k), k \in N^i$. The multiple paths computed between node i and destination j are called the *logical shortest multipath*, denoted by LSM^i_j, and is such that at least one of the paths in it has the minimal logical distance for j. Jn JARS, each node maintains up to x LSMs to each destination.

Given that each node must run JARS for each destination, we focus on the operation of node i's computation of LSMs for a destination j. Provided that each node maintains up to x LSMs for destination j, node i may receive and record x values of LD^i_{jk} from each neighbor k; node i also reports to its neighbors the logical distances of the x LSMs from itself to destination j, of which the minimal value is also used as the feasible logical distance FLD^i_j of node i. When a node is powered up, FLD is set to ∞, and all the other entries are set to empty. For destination j we have $LD^j_j = 0$, $FLD^j_j = 0$, and $LD^k_{jj} = 0, \forall k \in N^j$. We also assume that node i knows the transmission fraction of each outgoing link $TF_{(i,k)}, k \in N^i$.

When node i receives an input event at time t, node i behaves in one of three possible ways: 1). Node i remains idle and all distance estimates are left unchanged; 2). Node i receives LD^k_j from neighbor k, updates the estimates LD^i_{jk} and leaves all other estimates unchanged; 3). Node i updates $S^i_j(t)$ and $FLD^i_j(t)$ for destination j based on the following equations:

$$S_j^i(t) = \{k | LD_{jk}^i(t) < FLD_j^i(t),\ k \in N^i\} \tag{1}$$

and updates its feasible logical distance by

$$FLD_j^i(t) = min\left(LD_{jk}^i(t), \frac{1}{TF_{(i,k)}}\right) \tag{2}$$

for all LD_j^k reported by each neighbor k and over all neighbors in N^i.

Then node i re-computes the logical distance of each LSM maintained for j (up to x LSMs), and sends neighbors updates if any change occurs; otherwise leaves all other estimates unchanged.

The aggregate of the routing entries for destination j maintained at each node forms a directed graph rooted at j, which is a subgraph of network G and denoted by SG_j that includes links $\{l_{i,k} | k \in S_j^i \text{ for } \forall i \in V\}$. If routing converges correctly, SG_j is a directed acyclic graph (DAG) in which each node can have multiple successors for node j.

Although multiple SG_j can exist for destination j in a given network, JARS constructs SG_j in a way that the path with shortest logical distance for destination j is always maintained (by Eq. (1) and (2)), and as such makes SG_j an optimal successor graph.

Similar to the distributed Bellman-Ford (DBF) algorithm [18], JARS uses distance vectors to communicate logical distances only amongst neighboring nodes, and therefore avoids expensive routing overhead caused by disseminating link-state information throughout the network. JARS does not require each node to maintain complete network state and provides loop-free routing due to the using of Eqs. (1) and (2). The proof that using Eq. (1) to change successor sets cannot lead to loops is presented in [19].

4.6 Transmission fraction propagation and deduction

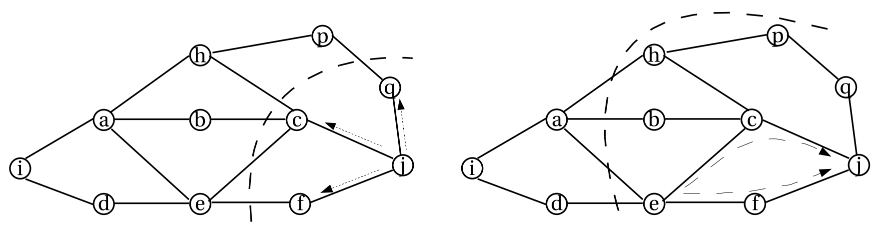
The distance vector reporting a path p for destination j by neighbor k is a tuple of $\{j, LD_j^k, TF_{(i,k)}\}$, in which LD_j^k is the logical distance for p, and $TF_{(i,k)}$ is the transmission fraction for adjacent link (i, k). For each LSM p computed for destination j, besides the logical distance, the raw transmission fraction of adjacent link $TF_{(i,k)}$ must also be maintained, because we need $TF_{(i,k)}$ to verify whether p can be a feasible path when a request to forward traffic arrives.

Assume that the minimal logical distance reported by neighbor k for destination j at node i is $\widetilde{LD}_{jk}^i$, and that the current feasible logical distance for j at node i is FLD_j^i. According to Eq. (1), path $l_{i,k} \circ p$ (operator $\circ$ is used to concatenate two paths or links) is now considered as a candidate path for j if $\widetilde{LD}_{jk}^i < FLD_j^i$ (*i.e.*, $k \in S_j^i$).

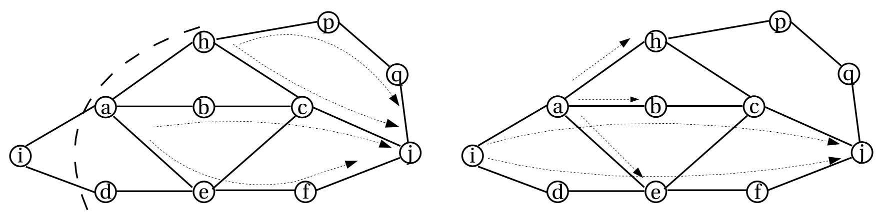
Path $l_{i,k} \circ p$ can be upgraded to a LSM if it has a smaller logical distance than the current feasible logical distance.In mobile scenarios, it may be the case that node i is unable to find a neighbor k that has reported a logical distance that is smaller than the feasible logical distance (FLD_j^i) maintained by node i at the time. JARS uses diffusing computations to coordinate node i with all upstream nodes that use node i in their LSMs for destination j to update the corresponding logical distance and feasible logical distance. The details of diffusing computations can be found in [20].

4.7 Example

Figure 4 demonstrates how nodes that run JARS to deduce their LSMs for the destination j without knowing global network state. Each node is labeled with (LD_j^i, FLD_j^i), i.e., its shortest logical distance and feasible logical distance for j; and each link is labeled with the associated transmission fraction.



(a) Computation of $TF_{qj}, TF_{cj}, TF_{fj}$ (b) Compute LSMs at node e



(c) Compute LSMs at node a, h (d) Compute LSMs at node i, a

Fig. 4. Routing example

In Figure 4(a), node j calculates the transmission fraction for link $\{(f,j),(c,j),(q,j)\}$, and sends the corresponding results through neighbor updates. After receiving the broadcast requests from j on the control channel, nodes $\{f,c,q\}$ switch one of its radio interfaces to the specified data channel to receive the neighbor update. Then nodes $\{f,c,q\}$ will calculate the transmission fractions for links $\{(e,f),(h,c),(b,c),(e,c),(p,q)\}$, and send the neighbor updates to their upstream nodes. In Figure 4(b), node e selects the optimal path to node j, and nodes $\{e,b,h,p\}$ will calculate the transmission fraction for links $\{(e,d),(e,a),(b,a),(h,a),(p,h)\}$. In Figure 4(c), node h chooses between paths $(h \rightarrow p \rightarrow q \rightarrow j)$ and $(h \rightarrow c \rightarrow j)$, node a chooses between paths $(a \rightarrow b \rightarrow c \rightarrow j)$ and $(a \rightarrow e \rightarrow f \rightarrow j)$ (we assume path $(e \rightarrow f \rightarrow j)$ is chosen over path $(e \rightarrow c \rightarrow j)$). This process continues until each node obtains the optimal path to the destination, as Figure 4(d) shows. Please notice that the schedule is formed *in sequence* along the routing path from the destination to the source. Descendent nodes will excludes the schedule of ascendent nodes which is indicated in $TS(c,t)$. Through this approach, the schedule and channel assignment along a specific routing path is compatible, while the schedules among different LSMs maybe in conflict. That is the reason why we just allow one LSM to be chosen each time. After the routes are established, the future neighbor update messages are sent through the existing transmission schedule and channel assignment as unicast packets, e.g. when j updates the TF_{qj}, it will just send a neighbor update message to q. Other nodes that do not use q in their LSMs to j will not receive the TF_{qj}.

5 Numerical Analysis

In this section, we numerically analyze the approximate MAC layer throughput and the complexity of routing layer protocol of JARS.

5.1 Approximate throughput analysis

To simplify the analysis, we consider a fully-connected network topology with N nodes. All links are bidirectional or symmetrical. Given that JARS increases the spatial reuse of the system through the distributed link scheduling in the two-hop range, a fully-connected network is the worst case scenario in terms of interference, contention or spatial reuse. Therefore, the throughput of JARS for a fully-connected network with N nodes is a lower bound of the throughput of JARS for a general topology where the number of nodes in a two-hop neighborhood is N. The channels are assumed to be error free and have no capture effects. Transmissions overlapped on the same channel at a receiver leads to a collision and no packets involved in it can be received correctly by the receiver.

We assume that traffic arrives at each node according to Poisson process with average arrival rate λ requests per slot. Therefore, the total traffic load is denoted by $G = N\lambda$.

We consider variable-length flow and assume that, on the average, it takes δ slots to send all the data packets in a flow, i.e., the average flow length (AFL) is δ slots. We also assume that the flow length is geometrically distributed, which implies that the probability that a flow ends at the end of a transmission slot is $q = 1/\delta$.

The system can be fully described by one state variable X, the number of communication pairs at time t. When $X = k$, $2k$ nodes are involved in data transmissions, while $N - 2k$ nodes are idle. The maximum number of communication pairs is bounded by the number of nodes and the number of data channels: $X \in \{0, 1, , \ldots, min(\lfloor \frac{N}{2} \rfloor, M - 1)\}$.

We model the evolvement of the system as a discrete-time Markov chain, where each state of the Markov chain can transit to any state. A transition may occur when new communication pairs are formed or existing transfers end. Let π_k denote the stationary probability that the system is in state k.

For the system is in state k, we denote the probability that n senders end their flows during a frame as $T_k^{(n)} = \binom{k}{n} q^n (1-q)^{k-n}, \quad 0 \le n \le k$.

We condition on the number of senders ending their flows in a frame (n) to calculate the transition probabilities. For the transition from state k in frame f to state l in frame $f+1$, at least $\hat{n} = max(0, k-l)$ nodes will end their flows in frame f. Therefore, $\hat{n} \le n \le k$, and $s = l - (k-n)$ new transmission pairs will be formed. We denote the probability that there is i new agreements made when the state is k as $\theta_k^{(i)}$. The transition probability from state k to state l is $P_{kl} = \sum_{n=\hat{n}}^{k} T_k^{(n)} \theta_k^{(s)}$.

Through solving the global balance equation: $\pi_l = \sum_{k=0}^{min(\lfloor \frac{N}{2} \rfloor, M-1)} \pi_k P_{kl}$.

with the boundary condition: $\sum_{l=0}^{min(\lfloor \frac{N}{2} \rfloor, M-1)} \pi_l = 1$.

The average channel utilization per channel can be obtained as $\rho = \frac{\sum_{i \in X_t} i\pi_i}{M}$.

The system throughput is $S = (M-1)C\rho$, where C is the channel capacity.

Since JARS adapts different channel assignment and link scheduling strategies according to network layer packet types, we analyze the $\theta_k^{(i)}$ of broadcast and unicast transmissions separately. We assume there exists only one type of traffic each time.

5.2 Broadcast transmission

Since all broadcast requests are sent on the control channel, at most one broadcast reservation can be made each time. Given an idle node contends for a slot with probability $p_a = 1 - e^{-\lambda}$, we can get the probability of making a successful broadcast reservation over the control channel:

$$\theta_k^{(i)} = \begin{cases} (N-2k)p_a(1-p_a)^{N-2k-1} & \text{if } i = 1; \\ 1-\theta_k^{(1)}, & \text{if } i = 0; \\ 0, & \text{otherwise.} \end{cases}$$

5.3 Unicast transmission

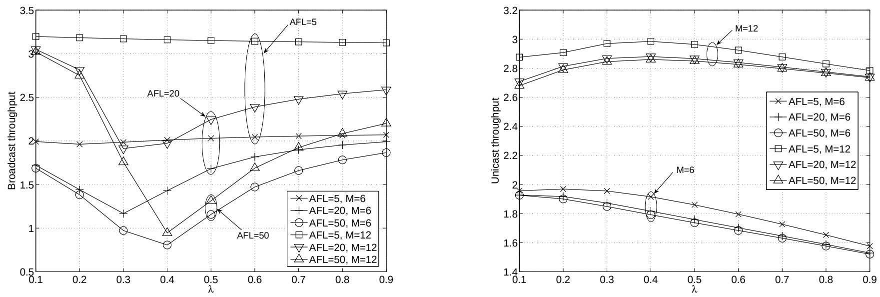
For unicast transmissions, we assume each node randomly chooses among $(M-1)$ data channels, then the average number of idle nodes on a data channel c when system is in state k is: $\overline{n_c} = \lceil \frac{N-2k}{M-1} \rceil$.

We can obtain the probability that there is successful reservation made on a specific data channel as $\eta = \overline{n_c} p_a(1-p_a)^{\overline{n_c}-1}$.

There can be up to $M-1$ successful reservations over all data channels:

$$\theta_k^{(i)} = \begin{cases} \binom{M-1}{i}\eta^i(1-\eta)^{M-1-i} & \text{if } 1 \le i \le M-1; \\ 1-\sum_{i=1}^{M-1}\theta_k^{(i)}, & \text{if } i = 0; \\ 0, & \text{otherwise.} \end{cases}$$

We set $N = 15$ and vary the traffic arrival rate from 0.1 to 0.9. We take $M = 6$ and $M = 12$ for example. The throughput comparisons for broadcast and unicast transmissions under different AFLs and traffic loads are shown in Figure 5(a) and Figure 5(b). We can find broadcast transmissions may experience throughput degradations when the AFLs are long. This is because



(a) Broadcast throughput (b) Unicast throughput

Fig. 5. Throughput analysis

the stationary probability of the communication pairs is dependent on two factors: 1). the successful transmission probability of the broadcast requests, which may be small for broadcast transmissions since all nodes contend on the same control channel; 2). the probability that there are available data channels, which is also small when the length of the traffic flow is long. This corresponds to the scenario that a lot of nodes contending on the control channel to send the broadcast requests, even if one of them succeeds with low probability, it may find the channel it tries to reserve is occupied by the long broadcast traffic flow. While for unicast transmissions, since transmission reservations are balanced throughout data channels, although its throughput also degrades with the increase of AFLs, it can still sustain a stable system throughput over all traffic loads. We note that is exactly the reason why JARS differentiates the broadcast and unicast transmissions.

5.4 Complexity analysis of routing protocol

We model the network as a directed graph $G = \{V, L\}$, where V is the set of nodes and L is the set of links interconnecting the nodes. N and E are the cardinalities of V and L, i.e., $N = |V|$ and $E = |L|$, respectively. For each node i runs JARS, the space complexity is $O(x|N_i|N + xN) = O(x|N_i|N)$, where N_i is the number of neighbors of node i, because the main routing table and each neighbor table have $O(N)$ entries, and each entry can keep up to x routes for each destination. The computation complexity of transmission fraction calculation algorithm is $O(D|N_i^2|^2)$, where N_i^2 is the number of nodes in the two-hop range of i, because in each time slot, the number of links to be scheduled is $O(|N_i^2|^2)$, and there are D time slots. The total time taken for a node to process distance vectors regarding a particular destination is $O(xD|N_i^2|)$.

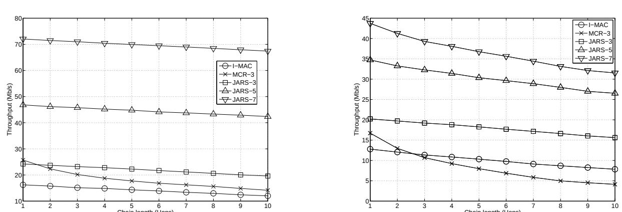
6 Performance Evaluation

6.1 Simulation settings

We implemented JARS under Qualnet [21]. We assume each node has four radio interfaces. The physical layer transmission rate of each channel is 54 Mbps. The transmit power is 16dBm. The receive threshold for 54Mbps data rate is -63dBm, the related transmission range is around 80m. We set the path loss factor $\alpha = 4$. The packet length used is 1024 bytes. Each node maintains up to three LSMs ($x = 3$). The neighbor updates are sent at the interval of 1 second. The duration of the simulation is 100 seconds. The simulations are repeated with ten different seeds to average the results for each scenario.

We vary the number of data channels to evaluate the capability of exploiting the channel diversity (designated as 'JARS-M', where is M is the number of channels).

In order to illustrate the performance gain due to the full-duplex radio and distributed scheduling, we first evaluate the performance of JARS in a simple chain topology. 11 nodes form a chain of 10 hops and direct communication is possible only between adjacent nodes on the chain. We set up a CBR/TCP flow over the chain and the flow length varies from 1 to 10 hops. Static routing is used which allows us to compare the performance without the influence of routing protocols. For CBR traffic, the traffic source continuously sends out data at the maximum possible rate so as to saturate the channel. We compare the performance of JARS with 1). 'I-MAC', for a specific chain length, there is an optimal channel assignment and transmission scheduling that could allow the maximal number of simultaneous transmissions; 2). MCR [14], it includes a link layer protocol implemented over 802.11 DCF and a routing metric for multi-channel multi-interface networks, which is incorporated into an on-demand routing protocol. We assume for MCR, each node has three channels (designated as 'MCR-3').



(a) CBR flows (b) TCP flows

Fig. 6. Throughput of chain topology

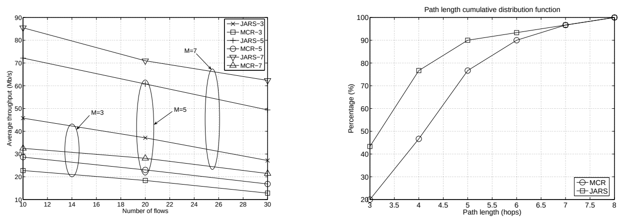
The system throughput comparisons for CBR and TCP traffic are shown in Figure 6(a) and Figure 6(b), respectively. Holland et al. [22] has shown that the throughput of CBR and TCP flows which are transmitted using 802.11

DCF will degrade rapidly when the number of hops along a chain increases. These are due to two reasons: 1). Intermediate nodes of the chain can not send and receive at the same time; 2). The transmissions on a hop will inhibit other transmissions on other hops.

JARS relaxes the node radio constraint through using full-duplex radios, which can be shown through the comparison with I-MAC. The lack of spatial reuse is addressed through the distributed transmission scheduling and channel assignment. Since JARS operates over a scheduling-based MAC protocol, when it has the same number of data channels with MCR, it outperforms MCR significantly under high traffic loads (the comparison between 'JARS-3' and 'MCR-3'). In summary, JARS reduces the throughput degradation effect of the chain topology, and can efficiently utilize the channel diversity of multichannel multi-radio system.

6.3 Random Topology

In order to illustrate performance improvement which is due to the joint optimization of MAC and network layer, we generate 10 topologies with 60 nodes uniformly distributed across a 800×800 square meters area. We compare the performance of JARS and MCR under following scenarios: 1). We set up multiple multihop CBR flows that are uniformly distributed across the network. We vary the number of CBR flows and channels. The average throughput comparison is shown in Figure 7(a). 2). We set $M = 6$ for both JARS and MCR, and compare the average path length. As Figure 7(b) shows, JARS reduces the average path length by one hop.



(a) Average throughput comparison

(b) Path length cumulative distribution function

Fig. 7. Random topology

The performance improvement of JARS over MCR not only comes from the underlying schedule-based MAC protocol, but also because that JARS incorporates the transmission scheduling and channel assignment information into the routing metric calculation, the optimal path is chosen based on all possible channel assignment and transmission scheduling combinations. While in MCR, the route is chosen based on the existing channel assignment and packet-based transmission scheduling (802.11 DCF), which is far from optimal.

Finally, in order to illustrate that JARS could dynamically assign the channels between the unicast and broadcast traffic when the traffic patterns changes, we compare the following two scenarios: 1). We first select 20 nodes as the broadcast sources, then add 10 CBR flows into the network; 2). We first initiate 10 CBR flows, then 20 nodes begin to periodically send broadcast packets. The packet length for both traffic is 1024 bytes. The traffic arrival interval of broadcast traffic is 0.05s. For CBR traffic, the traffic source continuously sends out data at the maximum possible rate so as to saturate the channel. The traffic sources and destinations are all randomly selected. The total number of channels is 3. The throughput change of different traffic types is shown in Table 1. We can find in both scenarios, JARS could allocate the bandwidth between different traffic flows. Since unicast transmissions will exclude the schedule and channel assignment of broadcast packets, which is included in $TS(c, t)$, broadcast transmissions have priorities over the unicast transmissions. The throughput decrease of broadcasts is less than unicasts.

Table 1

Throughput of dynamic traffic adaption (Mbps)

Scenario 1	Broadcast (without unicast)	Broadcast (with unicast)	Unicast
	3.11	2.54	33.75
Scenario 2	Unicast (without broadcast)	Unicast (with broadcast)	Broadcast
	41.34	32.29	2.51

7 Discussion

Currently JARS does not consider explicitly how to adapt the channel assignment and schedule strategies to the dynamic change of traffic patterns. This is due to two reasons: 1). The traffic patterns between each source and destination pair change frequently and are difficult to acquire; 2). Since there are no predetermined routing paths, how to split the traffic among all paths between a source and a destination is unknown. How to optimally allocate the traffic across different paths/links requires global information and is beyond the scope of this distributed scheme. We leave this part for future work. In JARS, we adapt a schedule-based MAC instead of a contention-based MAC because it can sustain high throughput under high traffic loads, and can potentially provide Quality-of-Service (Qos) support for real-time applications. The additional hardware requirements for the time synchronization is the cost of the performance improvements. The accuracy of time synchronization will partially influence the performance of JARS. We did not discuss this problem due to the page limits. More detailed discussion can be found in [23] [24].

8 Conclusion

In this paper, we propose a novel distributed link-layer scheduling and routing optimization approach for multi-channel multi-radio ad hoc networks. It adapts a pseudo full-duplex system to efficiently utilize the channel diversity of multi-channel multi-radio system. Routing selection is made based on the efficiency of underlying link layer scheduling and channel assignment schemes. Simulation results have shown that JARS increases the system performance significantly by decomposing the traffic over different channels and different time.

References

[1] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks", IEEE Trans. on Inf. Theory, vol. 46, pp. 388–404, March 2000.

[2] P. Kyasanur and N. Vaidya, "Capacity of Multi-channel Wireless Networks: Impact of Number of Channels and Interfaces", in ACM MobiCom, 2005, pp. 43–57.

[3] A. Raniwala and T. chker Chiueh, "Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network", in IEEE INFOCOM, March 2005.

[4] A. Raniwala, K. Gopalan, and T. cker Chiueh, "Centralized Algorithms for Multi-channel Wireless Mesh Networks", ACM SIGMOBILE Mobile Computing and Communications Review, vol. 8, no. 2, pp. 50–65, 2004.

[5] P. Kyasanur and N. H. Vaidya, "Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks", in IEEE WCNC, 2005.

[6] M. Kodialam and T. Nandagopal, "Characterizing Achievable Rates in Multi-hop Wireless Networks: The Joint Routing and Scheduling Problem", in Proceedings of ACM MobiCom, 2003, pp. 42–54.

[7] J. Zhang, H. Wu, Q. Zhang, and B. Li, "Joint Routing and Scheduling in Multi-radio Multi-channel Multi-hop Wireless Networks", in Proceedings of Broadnets, 2005, pp. 631–640.

[8] M. Alicherry, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh networks", in Proceedings of ACM MobiCom, 2005.

[9] M. Kodialam, T. Nandagopal, "Characterizing the Capacity Region in Multi-radio, Multi-channel Wireless Mesh Networks", in Proceedings of ACM MobiCom, 2005, pp. 73–87.

[10] X. Meng, K. Tan, and Q. Zhang, "Joint Routing and Channel Assignment in Multi-radio Wireless Mesh Networks", in Proceedings of IEEE ICC, 2006.

[11] W.-H. Tam and Y.-C. Tseng, "Joint Multi-Channel Link Layer and Multi-Path Routing Design for Wireless Mesh Networks", in IEEE INFOCOM, 2007.

[12] H. Wu, F. Yang, K. Tan, J. Chen, Q. Zhang, and Z. Zhang, "Distributed Channel Assignment and Routing in Multi-radio Multi-channel Multi-hop Wireless Networks", Journal on Selected Areas in Communications, vol. 24, pp. 1972–1983, Nov 2006.

[13] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio Multi-Hop Wireless Mesh Networks", in ACM Mobicom, 2004.

[14] P. Kyasanur and N. H. Vaidya, "Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks", ACM SIGMOBILE Mobile Computing and Communications Review, vol. 10, no. 1, pp. 31–43, January 2006.

[15] S. Yin, Y. Xiong, Q. Zhang, and X. Lin, "Prediction-based Routing for Real Time Communications in Wireless Multi-hop Networks", in Proceedings of QSHINE06, Waterloo, Canada, August, 2006.

[16] P. Bahl and R. Chandra and J. Dunagan, "SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks", in ACM Mobicom, 2004.

[17] J. Mo, H. So, and J. Walrand, "Comparison of Multi-channel MAC Protocols", in International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, October, 2005.

[18] D. Bertsekas and R. Gallager, Data Networks" - 2nd Edition. Prentice-Hall, 1992.

[19] Z. Li and J. J. Garcia-Luna-Aceves, "A Distributed Approach For Multi-Constrained Path Selection And Routing Optimization", in Proceedings of QSHINE06, Waterloo, Canada, August, 2006.

[20] J. J. Garcia-Lunes-Aceves, "Loop-free Routing Using Diffusing Computations", IEEE/ACM Trans. Netw., vol. 1, no. 1, pp. 130–141, 1993.

[21] Qualnet Simulator, "Scalable Network Technologies", http://www.scalable-networks.com/.

[22] G. Holland and N. Vaidya, "Analysis of TCP Performance over Mobile Ad Hoc Networks", in Proceedings of ACM Mobicom, 1999, pp. 219–230.

[23] Carlos H. Rentel, "Network Time Synchronization and Code-based Scheduling for Wireless Ad Hoc Networks", Ph.D. Thesis, Carleton University, Ontario, Canada, January 2006.

[24] K. Romer and E. Zurich, "Time Synchronization in Ad Hoc Networks", in Proceedings of ACM Mobihoc, 2001.