Joint Non-bifurcated Routing and Scheduling in Wireless Grid Mesh Networks

Abdullah-Al Mahmood and Ehab S. Elmallah
Department of Computing Science
University of Alberta
Edmonton, T6G 2E8, Canada
E-Mail: {amahmood,ehab}@cs.ualberta.ca

***Abstract*—In this paper we consider multi-hop wireless mesh networks intended to provide Internet connectivity to both end users and hotspots. In such networks mechanisms for provisioning QoS for delay sensitive flows arise as an important topic. In this context, we focus on the non-bifurcated (single path) routing of such flows as a means of allowing all packets in any flow to receive uniformly controlled treatment in each node along the path, thus simplifying routing and management of flows requiring QoS guarantee. In particular, we consider non-bifurcated routing in the class of wireless grid mesh networks (WGMNs). We formalize the problem of joint routing and scheduling of flows that can be best served by non-bifurcated routes as an optimization problem. We then discuss the advantages of solving the problem by a strategy that attempts to route and schedule pairs of flows at each step. Subsequently we propose a dynamic programming algorithm to compute such routes. We evaluate the performance of the proposed algorithm both analytically and by simulation. The experimental results show that the proposed algorithm performs better than two other commonly used competing strategies.**

Keywords: wireless mesh networks, fixed broadband wireless access, non-bifurcated routing, dynamic programming

I. Introduction

Broadband access over wireless networks is currently gaining worldwide attention as a means of constructing low cost flexible networks capable of serving the growing demand for Internet access. Wireless mesh networks (WMNs) based on the IEEE WirlessMAN/HUMAN 802.16 standards support broadband wireless access (BWA) in two important deployment scenarios: *last mile* networks for connecting subscriber stations to backbone networks, and *backhaul* networks for forwarding aggregated traffic from hotspots and the edge of cellular networks to backbone networks. Last mile networks utilize the *point-to-multipoint* (PMP) mode of operation, whereas backhaul networks utilize the multi-hop *mesh* mode of operation.

In this paper we consider the joint routing and scheduling problem for traffic in multi-hop mesh networks. Our goal is to develop schemes that achieve high throughput values while simplifying monitoring and management of flows requiring QoS guarantees (e.g., delay sensitive traffic and delay-jitter sensitive traffic). Such goals can be approached by exploring single path (non-bifurcated) routing protocols. Our work here contributes to this direction by considering deployment scenarios where the topology of the network is (or can be approximated by) a grid. We refer to such networks as wireless grid mesh networks (WGMNs).

Many existing results in the literature on routing and scheduling in multi-hop WMNs are related to our work here. Below we mention some of such results.

In [1] and [2] the authors develop distributed channel assignment and routing algorithms for multi-hop multi-channel WMNs where each channel forms a single collision domain. In [3] the authors develop a centralized algorithm for maximizing throughput while achieving fair allocation in multi-hop WMNs. In [4] the authors develop a centralized algorithm for fair scheduling of data transmissions in multi-hop WMNs. The work in [5] and [6] analyze the capacity of multi-hop multi-channel multi-radio WMNs. Similar to our work here, all routers are assumed to transmit data in well-defined timeslots. The authors develop schemes for solving the underlying joint routing and scheduling cross-layer optimization problem. However, unlike our work, the proposed schemes in [5] and [6] employ multi-path routing where a single flow is allowed to split among multiple paths before reaching the destination.

In comparison with the above work, our work here concerns the joint non-bifurcated routing and scheduling in mesh networks. We remark that the advantages of non-bifurcated routing have been discussed in [7] for ad-hoc networks and in [8] for sensor networks. Forwarding all packets of a given traffic flow over a single path enables uniform treatment of the packets in each intermediate node which promotes efficient handling of flows requiring QoS guarantees and enables simplified network monitoring and management. In the context of the IEEE 802.16 WMNs, flows that can benefit from non-bifurcated routing include Unsolicited Grant Service (UGS) flows and Real-time Polling Service (rtPS) flows. In a previous work [9], we considered a non-bifurcated routing problem in arbitrary networks where routers use a CSMA/CA protocol.

Here we explore a joint non-bifurcated routing and scheduling problem in the important class of grid networks that is often discussed in the wireless networking literature. Our main contribution is a novel routing and scheduling algorithm that strives to allocate resources at each step for a pair of flow demands simultaneously in grid networks where routers use a TDMA protocol. Some advantages of this approach are discussed in section IV.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7101
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7101

II. System Model

As in the work of [5] and [6] in this paper we consider WMNs with fixed mesh routers where one of the mesh routers acts as a gateway to the wired Internet. The mesh routers form a *backbone network* for backhauling traffic from individual subscribers and hotspot access points. We assume that routers are synchronized in time so as to allow data transfers between adjacent routers in well-defined time-slots. End users connect to the backbone network and the gateway through their nearest mesh router. Such backbone WMNs are often envisioned to utilize multiple channels and multiple radios at each mesh router. Harnessing the available resources, however, often hinges on the effectiveness of utilizing the bandwidth in each available channel. In this paper we focus on the single channel case as a fundamental and important subproblem of the more general multi-channel case.

We assume that the local communication between subscribers or access points and their nearest mesh router is carried over a secondary wireless channel that is orthogonal to the primary wireless channel used by the backbone WMN. In this paper we assume that the backbone network operates in the multi-hopping mesh mode. Communication between end users and their closest mesh router may utilize the point-to-multipoint mode.

We consider a WGMN of width W units and height H units consisting of $(W+1) \times (H+1)$ mesh routers. Each router is assumed to be unit distance apart from its nearest neighbors. Although the grid can be rectangular, the grid cells are assumed to be square and routers are located at the corners of each square. For simplicity, the gateway is assumed to be located at the bottom left corner of the grid. Each router is assigned an (x,y)-coordinate (i,j) where $i \in [0,W]$, and $j \in [0,H]$, and the gateway is located at coordinates $(0,0)$. We assume that all routers have the same transmission range R_T and the same interference range R_I ($R_I \geq R_T$). The transmission range R_T is adjusted so that each mesh router can directly communicate with its closest neighbors only. In such a WGMN transmission links coincide with grid lines. A transmission link between routers (i_1,j_1) and (i_2,j_2) is denoted $\langle (i_1,j_1),(i_2,j_2)\rangle$.

The interference between routers is determined by R_I. At each time slot we require that all routers within R_I distance of a sender or a receiver of a transmission to be inactive. This requirement allows bi-directional data transfers over each link since the role of a sender and a receiver can be exchanged without affecting the link interference relations. That is, the computed routes can be used for both uplink and downlink communication with the gateway. For example, in Fig. 1(a), assuming $R_I = R_T$, transmissions over link A interfere not only with transmissions over links b, and B, but also with transmissions over links c and C.

For our purpose, we define the *cross-interference* (CI) function of two sets of links as the number of link-pairs (one link from each set) that can interfere with each other under this interference model. We denote this function by $f(\cdot)$. For example, in Fig. 1(a), $f(\{A\},\{a,B,b,C,c\}) = 5$.

In a time-slotted system, a *frame* consists of a number of consecutive time slots. A *schedule* specifies transmission activity (namely transmission, reception, or inactivity) for each link in each time slot of a frame. The number of time slots in a frame is referred to as the *schedule length*. The schedule length and the duration of a time slot is determined from packet size and the channel capacity. The maximum amount of data that can be transmitted over a single link during one time-slot is considered to be a unit of flow. Network traffic in our model is measured using such flow units.

In a WGMN each end user can request bandwidth from its nearest mesh router (to which the end user connects) for different applications. We assume a centralized resource management scheme where the mesh routers periodically forward all such requests to a central computing facility (e.g., the gateway), and the facility computes routes, and bandwidth allocations in the form of a schedule. The computed results are conveyed back to the mesh routers. Such centralized resource management scheme aims at analyzing the maximum achievable throughput that the network can deliver using simpler distributed algorithms.

III. Problem Formulation

In this paper we consider the following throughput maximization problem: given a set of traffic flows requested by end users and traffic aggregation points, we seek to compute the maximum amount of flows that can be jointly routed and scheduled under the *route indivisibility* constraints, *interference* constraints, and *schedule length* constraint described below. The inputs and outputs of the problem are as follows:

Inputs:

- $T_{\max}$: The maximum allowable length of the sought after schedule.
- $D(i,j)$: A set $\{D_t(i,j) : 1 \leq t \leq |D(i,j)|\}$ specifying traffic demands corresponding to each flow t at router (i,j). Each demand value $D_t(i,j)$ is an integer (using our flow unit).

Outputs:

- $S(i,j)$: A set specifying the traffic demands accepted for routing at router (i,j) (i.e., $S(i,j) \subseteq D(i,j)$).
- T : A table with at most $T_{\max}$ rows that stores the computed schedule with one row for each time slot. Row i, $i \in [1,T_{\max}]$, corresponds to the transmission activities during the i-th time-slot in each frame. Specifically the i-th row is a list of pairs of values where each pair specifies a flow identification number and a link along which transmission of the corresponding flow takes place.

The set of accepted flows $\{S(i,j)|i \in [0,W], j \in [0,H]\}$ must satisfy the following constraints:

Flow Conservation Constraint: For any non-gateway router (i,j), the sum of outgoing flows must equal the sum of incoming flows from other routers and flows accepted from end users.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7101
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7101

Flow Indivisibility Constraint: Any accepted traffic demand $D_t(i,j) \in S(i,j)$ must be assigned an unsplittable route to the gateway.

Interference Constraint: For each row in T containing a set s of transmissions, the transmissions must be pairwise cross-interference free. i.e., any two entries $[\mathrm{id}_1, e_1]$, $[\mathrm{id}_2, e_2] \in s$ satisfy $f(\{e_1\},\{e_2\}) = 0$.

Schedule Length Constraint: The number of rows in table $T \leq T_{\max}$.

We remark that various throughput maximization problems in multi-hop wireless networks have been shown to be NP-hard (e.g., see the results in [10], [11]). Such results justify the need to develop effective heuristic algorithms as pursued in our work.

IV. Proposed Solution Approach

In this section we highlight a number of insights that lie behind the design of our proposed algorithm.

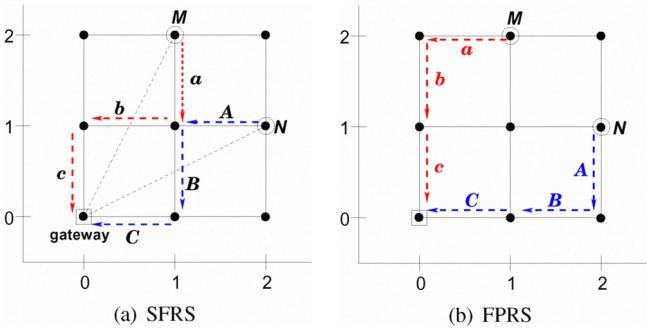
First, we remark that a core subproblem in our joint routing and scheduling optimization problem can be stated as follows: given a schedule T that specifies the routing and timing information for a given number of flows, and a new flow that is required to be routed along with the existing flows in T, can such a flow be accommodated without perturbing T? We call this problem *single flow joint routing and scheduling* (SFRS, for short) problem. We remark that the SFRS problem for arbitrary interference ranges and arbitrary routes appears to be a computationally demanding combinatorial problem. Nevertheless the availability of an efficient solution to the SFRS problem can be used by an iterative algorithm to construct a suboptimal solution to the optimization problem described in section III.

Second, a generalized version of the SFRS problem calls for finding whether a given pair of flows can be added to an existing schedule T. We call this generalized problem a *flow-pair joint routing and scheduling* (FPRS, for short) problem. An important insight that underlies our work in this paper is that using an effective solution to the FPRS problem can achieve a substantial improvement over the use of effective solution to the SFRS problem, as illustrated by the following example.

Example. Fig. 1(a) illustrates a 2×2 WGMN with two source routers M and N, and a gateway. Each router has a flow of one unit to send to the gateway. We require that $T_{\max} = 4$. For simplicity, we assume that $R_T = R_I$.

A solution to the SFRS problem may return path $A-B-C$ or $a-b-c$. In this case, one may verify that only one flow is admissible under the given schedule length constraint. On the other hand, Fig. 1(b) and table I illustrate that a solution to the FPRS problem can admit both flows in a schedule of length 4. This is a two-fold improvement in the throughput. ■

Third, an effective solution to the FPRS problem can not only achieve higher values of the objective function but also yield schedules of shorter length, as illustrated in the following example.



(a) SFRS (b) FPRS

Fig. 1. SFRS vs. FPRS

Time-slot	Transmissions
1	$[\mathrm{id}_1, a]$, $[\mathrm{id}_2, A]$
2	$[\mathrm{id}_1, b]$, $[\mathrm{id}_2, B]$
3	$[\mathrm{id}_1, c]$
4	$[\mathrm{id}_2, C]$

TABLE I
Transmission schedule T

Example. In Fig. 1(a), the iterated use of an algorithm for solving SFRS may compute the two routes $A-B-C$, and $a-b-c$. One may verify that a constructed schedule that utilizes these two routes requires 6 slots. On the other hand, the two routes in Fig. 1(b) that can be computed by an algorithm for solving FPRS problem can be scheduled in 4 slots. The example demonstrates a 33% gain in schedule length resulting from solving the FPRS problem. ■

Fourth, designing an effective solution to the FPRS problem entails (a) finding good candidate routes for serving the given pair of flows, and (b) determining whether the transmissions along the given pair of routes are schedulable along with existing reservations in input schedule T. For the routing part in (a), our proposed algorithm considers only routes with shortest rectilinear distance to the gateway. Such routes are likely to introduce minimal interference on the neighboring routers, and hence allow more flows to be served.

For the scheduling part in (b), our proposed algorithm assigns the most-utilized time-slot to transmission over a link without violating the interference constraint. The *most utilized first* (MUF) slot heuristic described above is likely to make the constructed schedule extensible.

Lastly, as a WGMN offers a potentially large number of shortest path routes between a router and the gateway, an effective algorithm to solve the FPRS problem should be capable of using this rich set of routes. Our proposed algorithm uses a dynamic programming approach where the set of available routes is examined in stages. In each stage, a route pair that is considered most extensible is maintained by the algorithm. We remark that a pair of partial routes with a small CI value has a good potential for time-slot reuse, and thus such pair of routes provides a good potential for being extensible. Consequently, the algorithm views a pair of routes (r_1, r_2) with the least cross-interference value $f(r_1, r_2)$ as being the most extensible pair of routes that should be maintained for

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7101
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7101

successive computations.

Example. In Fig. 1(a), the two routes illustrated have a CI value of 9 whereas in case of the routes in Fig. 1(b), the CI value is 3. Based on the use of the CI metric, the proposed algorithm discards the pair of routes in Fig. 1(a) in favor of keeping the pair of routes in Fig. 1(b). ■

V. The Algorithm

In this section we outline the main steps of a dynamic programming algorithm to solve the FPRS problem (function FPRS in Fig. 2). The function takes as input a schedule T, and the (x, y)-coordinates (x_M, y_M) and (x_N, y_N) of two routers M and N. Without loss of generality we assume that $x_M \leq x_N$, i.e., of the two routers, M can be considered the leftmost router. For simplicity of description, each router is assumed to be a source of a unit flow that needs to be routed to the gateway at $(0, 0)$. If the pair of flows sourced at routers M and N can be scheduled along with the existing flows in T without perturbing the existing slot assignments, then the function updates T by adding the computed routes and the corresponding slot assignments to serve the new flows.

The computations are done in stages. Graphically the stages correspond to moving a hypothetical vertical scanline from the leftmost column ($x = 0$) in the WGMN to the rightmost column corresponding to $x = x_N$. For a given position of the scanline at a particular x-coordinate x, the algorithm considers every pair of routers at the coordinates (x, y), and (x, y'). It is possible that $y = y'$. The algorithm decides whether the schedule T admits two flows sourced from these two particular routers to the gateway. If the answer is yes, then the algorithm maintains two routes r_1, r_2 with minimum cross-interference value $f(r_1, r_2)$. Computations of such pair of routes utilize routes previously computed by the algorithm.

Below we summarize the key data structures and functions utilized by the algorithm.

1) **Table A:** A is a three-dimensional table where the key of each entry corresponds to a triple (x, y_1, y_2) where x is the position of the vertical scanline, and (x, y_1) and (x, y_2) are two routers. The interpretation of the entry depends on the relative position of the scanline with respect to the x-coordinates of the routers M and N as follows:
   - If the scanline position $x \leq x_M$, then the value of $A[x, y_1, y_2]$ is a pair of routes from the two routers (x, y_1) and (x, y_2) respectively to the gateway along which a unit of flow can be transmitted without violating interference constraints.
   - If the scanline position $x > x_M$ then the value at $A[x, y_M, y_2]$ stores routes from routers M and (x, y_2) to the gateway. The properties of the routes are the same as those described above.
2) **Function *extend*:** This function takes two parameters. The first parameter is an entry in table A (say $A[x, y_1, y_2]$) and the second one is an ordered pair of grid edges (say e_1 and e_2). The function checks whether the edges e_1 and e_2 can be appended to the first and second routes at $A[x, y_1, y_2]$ respectively so that the transmissions over the extended routes can be scheduled along with existing transmissions in T. If so, the function returns the pair of extended routes.

Function FPRS (T, M, N)

Inputs: i) Coordinates (x_M, y_M) and (x_N, y_N) of the source routers M and N where $x_M \leq x_N$
ii) Schedule T

Outputs: i) A pair of non-bifurcated routes in $A[x_N, y_M, y_N]$ or null (in case of failure)
ii) Updated T if the route pair is schedulable

1. Gateway initialization:
   $A[0, 0, 0] \leftarrow \Big([(0, 0)], [(0, 0)]\Big)$
2. Leftmost scanline ($x = 0$) initialization:
   for ($i = 0$ to y_M, $j = 0$ to y_N, and $i + j > 0$) **do**
   case $i, j > 0$:
   $A[0, i, j] \leftarrow$ extend$(A[0, i-1, j-1],$ $\Big($Down$(0, i),$ Down$(0, j)\Big))$
   case $i = 0, j > 0$:
   $A[0, i, j] \leftarrow$ extend$(A[0, i, j-1], \Big(\emptyset,$ Down$(0, j)\Big))$
   case $i > 0, j = 0$:
   $A[0, i, j] \leftarrow$ extend$(A[0, i-1, j], \Big($Down$(0, i), \emptyset\Big))$
   end for
3. Bottom grid line initialization:
   for ($x = 1$ to x_N) **do**
   case $x \in [1, x_M]$:
   $A[x, 0, 0] \leftarrow$ extend$(A[x-1, 0, 0],$ $\Big($Left$(x, 0),$ Left$(x, 0)\Big))$
   case $x \in (x_M, x_N]$:
   $A[x, 0, 0] \leftarrow$ extend$(A[x-1, 0, 0], \Big(\emptyset,$ Left$(x, 0)\Big))$
   end for
4. Non-boundary entries:
   for ($x = 1$ to x_N) **do**
   4.1 **case** $x \in [1, x_M]$:
   for ($i = 0$ to y_M, $j = 0$ to y_N, and $i + j > 0$) **do**
   $A[x, i, j] \leftarrow$ best-pair $\Big\{$
   extend$(A[x-1, i, j], \Big($Left$(x, i),$ Left$(x, j)\Big))$,
   extend$(A[x, i-1, j], \Big($Down$(x, i), \emptyset\Big))$,
   extend$(A[x, i, j-1], \Big(\emptyset,$ Down$(x, j)\Big))$,
   extend$(A[x, i-1, j-1],$ $\Big($Down$(x, i),$ Down$(x, j)\Big))\Big\}$
   end for
   4.2 **case** $x \in (x_M, x_N]$:
   for ($j = 1$ to y_N) **do**
   $A[x, i, j] \leftarrow$ best-pair $\Big\{$
   extend$(A[x-1, y_M, j], \Big(\emptyset,$ Left$(x, j)\Big))$,
   extend$(A[x, y_M, j-1], \Big(\emptyset,$ Down$(x, j)\Big))$ $\Big\}$
   end for
   end for
5. If $A[x_N, y_M, y_N]$ is not empty (i.e., the pair of routes is schedulable), then update T and return $A[x_N, y_M, y_N]$, else return null.

Fig. 2. Pseudo-code of function FPRS

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7101
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7101

Otherwise the function indicates failure. In case the first (or second) route need not be extended we set $e_1 = \emptyset$ (or $e_2 = \emptyset$ respectively).

3) **Function *best-pair*:** The best-pair function takes a list of route pairs as argument, computes the CI value of each route pair, and returns the route-pair with the minimum CI value.
4) **Functions *Down* and *Left*:** These two functions identifies links adjacent to a router that are horizontally to the left and vertically down with respect to the position of the router. In particular, Down(i, j) refers to the link $\langle (i, j-1), (i, j) \rangle$ and Left(i, j) refers to the link $\langle (i-1, j), (i, j) \rangle$.

The overall steps of the algorithm can be described as follows: Step 1 initializes the entry $A[0, 0, 0]$ associated with the gateway. Step 2 initializes boundary values $A[0, i, j]$ corresponding to the leftmost column. Step 3 initializes the boundary values $A[x, 0, 0]$ corresponding to the bottom row of the grid. Step 4.1 computes entries of the general form $A[x, i, j]$, $x > 0$. Step 4.2 deals with the case when the scanline has moved to the right of router M.

We remark that WGMNs are not expected to be large in size. However, they are expected to carry a large number of flows. We next show that the running time of our algorithm grows linearly with the number of carried flows.

Running Time. For a $W \times H$ WGMN a single run of function FPRS requires $O(WH^2)$ entries in table A to be filled. Each entry stores a pair of routes with $O(W + H)$ links. Each entry requires a constant number of routes to be examined for extension by testing schedulability and computing pairwise-conflicts. If the number of scheduled flows in T is f_s then the number of links in T that need to be tested for conflicts is $O(f_s(W + H))$. Computation of each entry therefore requires $O(f_s(W + H)^2)$ time. The total running time to compute a schedulable route pair is therefore $O(f_s WH^2(W + H)^2)$. If the grid is almost square, and the number of routers in the grid is n then W and H both are $\Theta(\sqrt{n})$. In that case the running time of the algorithm is $O(f_s n^{5/2})$.

VI. Route Selection

Given two routers M and N in a WGMN, each having a flow of a certain amount to communicate with the gateway, and a schedule T of previously scheduled flows in the network, function FPRS strives to find a pair of routes that serves the flows from M and N along with the existing flows in T without changing T's slot assignments. We call such pair of routes *feasible*.

An important feature of the solution approach adopted by function FPRS is the generation and maintenance of feasible pair of partial routes that enjoy relatively small CI value. As mentioned earlier, partial route pairs that have the small CI property are more likely to admit extensions to complete routes from each of M and N to the gateway. In this section we present key observations that collectively show that route pairs generated by the algorithm enjoy the small CI property. The presentation considers the computations done in three different phases as explained below.

Phase 1. The scanline ℓ is in the leftmost position $\ell = 0$. Here, the algorithm considers all possible choices of router positions y_1 and y_2 on the vertical line $x = 0$ (i.e., $y_1 \in [0, y_M]$ and $y_2 \in [0, y_N]$). For each such router, y_i, $i = 1$ or 2, the algorithm considers the unique route r_i made of vertical edges to the gateway. The algorithm keeps routes r_1 and r_2 in entry $A[\ell = 0, y_1, y_2]$ if they are feasible, otherwise the entry is null. Thus the algorithm stores a pair of feasible routes having the smallest possible CI value, if such pair exists.

Phase 2. The scanline $\ell \in [1, x_M]$. The algorithm computes entries of the form $A[\ell, y_1, y_2]$ for each possible setting of $y_1 \in [0, y_M]$ and $y_2 \in [0, y_N]$. For each such pair of values y_1 and y_2 the algorithm aims at computing two feasible partial routes r_1 and r_2, where r_1 connects router (ℓ, y_1) to the gateway, and r_2 connects router (ℓ, y_2) to the gateway. We write $r_1 = r'_1 + e_1$ (similarly, $r_2 = r'_2 + e_2$) where e_1 (respectively, e_2) is either a down link or a left link incident with node (ℓ, y_1) (respectively, (ℓ, y_2)). Routes r'_1 and r'_2 are the remaining parts of route r_1 and r_2, respectively, that have been computed in a previous step. Consequently, the pair (r'_1, r'_2) is assumed to enjoy the small CI property.

We say that edge e_1 lies above e_2 if $y_1 > y_2$, or if $y_1 = y_2$ and the y-coordinate of the other end vertex of e_1 is greater or equal to its counterpart of e_2. We draw the following remarks assuming that e_1 lies above e_2 (the argument is symmetric if e_2 lies above e_1):

2.a. The contribution of edges e_1 and e_2 to $f(r_1, r_2)$ is $f(e_1, r'_2) + f(e_2, r'_1)$.
2.b. In all cases, $f(e_1, r'_2) \leq 2$, and the exact value of $f(e_1, r'_2)$ is independent of the layout of r_2 in all cases except for one case (when $y_1 = y_2$).
2.c. For $f(e_2, r'_1)$, we have $f(e_2, r'_1) = 0$ if $y_1 \geq y_2 + 3$, independent of e_2 and r'_1.
2.d. We also observe that:
$f(e_2, r'_1) \leq 1$ if $y = y_2 + 2$, and
$f(e_2, r'_1) \leq 3$ if $y_1 = y_2$ or $y_2 + 1$.
In these cases the exact value depends on e_2.

Phase 3. The scanline ℓ moves to the right of router M, i.e., $\ell \in [x_M + 1, x_N]$. Here the algorithm utilizes entries of the form $A[\ell, y_M, y_2]$ where $y_2 \in [0, y_N]$. For each possible value of y_2, the algorithm aims at computing a pair of routes r_1 and r_2, and storing them in entry $A[\ell, y_M, y_2]$ where r_1 is a route from M to the gateway, and r_2 is a route from (ℓ, y_2) to the gateway. We write $r_2 = r'_2 + e$ where e is either a down link or a left link incident with router (ℓ, y_2), and r'_2 is the remaining part of r_2. We remark that

3.a. For all $\ell \geq x_M + 2$, $f(e, r_1) \leq 1$ and the exact value is independent of r_1.
3.b. For $\ell = x_M + 1$, $f(e, r_1) \leq 2$. The exact value depends on r_1 only when y_2 is sufficiently close to y_M.

Thus, although at each step the algorithm computes a feasible pair of routes (if such pair exists) with the minimum encountered CI value, this value may not be optimum in some

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7101
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7101

cases when the routes are required to be within the interference range of each other, as described in remarks 2.d and 3.b above. As the majority of pairs of routes that are candidates of being feasible are expected to be close to each other only in a limited number of locations, we conclude that all such pairs of routes computed by the algorithm possess a small CI value as desired.

VII. Experimental Results

In this section we evaluate the effectiveness of our algorithm with respect to the ability to (a) maximize network throughput, and (b) construct schedules of relatively short length. We compare its performance against two commonly used algorithms that route one flow in each step. The first algorithm uses a near straight line routing (SLR) heuristic to route each flow under consideration. The second algorithm forwards packets to the neighboring router closest in Eucledian space to the gateway (denoted the CGF algorithm below).

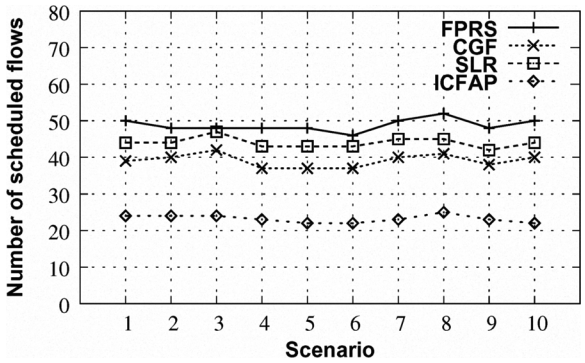


Fig. 3. Achieved throughput

We also compare performance against a third algorithm that does not require routers to be synchronized (i.e., does not use time-slots). The third algorithm, called the interference-constrained flow augmenting path (ICFAP) search algorithm, is described in [9]. The algorithm is implemented in Qualnet 3.9.5 and has been shown to deliver superior performance over the Dynamic Source Routing (DSR) ad-hoc routing algorithm. Its performance, however, is expected to be weaker than the performance of algorithms utilizing time-slotted transmissions.

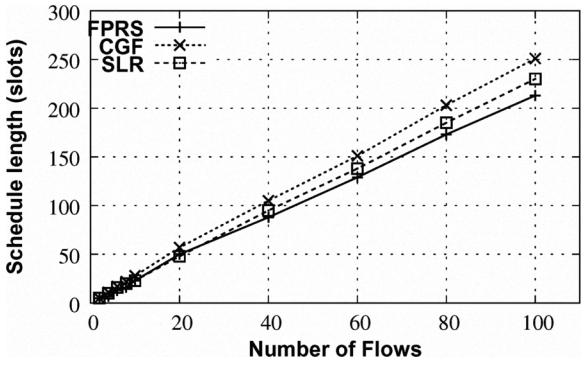


Fig. 4. Achieved schedule length

All test cases uses a 8×6 grid network with one gateway and 62 other mesh routers. The generated traffic demand has 60 randomly generated flows, each of unit value. Fig. 3 illustrates the achieved throughput (i.e., the number of admitted flows) in 10 different scenarios where each scenario corresponds to a random way of generating 60 flows by the 62 mesh routers, and we seek to construct a schedule of length $T_{\max} = 100$. As can be seen, our FPRS algorithm consistently outperforms other methods (e.g., by as much as 12% over SLR and 22% over CGF). Fig. 4 illustrates the effectiveness of the algorithm in constructing schedules of short length (e.g., SLR and CGF produces schedules that require 8% and 17% more slots respectively). Here we incrementally add a number of flows to a network so as to vary the total number of demands from 0 to 100. After each addition, we compute a schedule with minimum length that accommodates all flows.

VIII. Concluding Remarks

In this paper we consider a joint routing and scheduling problem for routing traffic in a multi-hop WMN deployed in a grid configuration. In such networks, forwarding all packets of a given traffic flow over a single path enables uniform treatment of all packets in each intermediate node as well as promotes simplified network monitoring and management. Motivated by the above advantages, the paper develops a novel routing and scheduling algorithm to solve the underlying cross-layer combinatorial optimization problem. The devised algorithm acquires its strength from dealing with the combinatorics of serving pairs of flows at each step. Our algorithm applies to traffic carried over a single wireless channel and works in a centralized way. For future research, we propose investigating both centralized and distributed algorithms for solving joint non-bifurcated routing and scheduling problems on multi-channel WMNs. Obtaining results in this direction complements existing routing results on WMNs that allow potentially unlimited splitting of flows within the network.

IX. Acknowledgments

This work is supported by NSERC Canada under grant number OGP 36899.

References

[1] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom '04: Proc.f the 10th annual intl. conf. on Mobile computing and networking.* New York, NY, USA: ACM Press, 2004, pp. 114–128.

[2] A. Raniwala and T.-c. Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," in *INFOCOM 2005. Proc. of the IEEE 24th Annual Joint Conf. of the IEEE Computer and Comm. Societies.*, vol. 3. IEEE, March 2005, pp. 2223–2234.

[3] Y. Bajerano, S.-J. Han, and A. Kumar, "Efficient load-balancing routing for wireless mesh networks," *Computer Networks*, vol. 51, no. 10, pp. 2450–2466, July 2007.

[4] N. Ben Salem and J.-P. Hubaux, "A fair scheduling for wireless mesh networks," in *The First IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[5] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," *IEEE Jnl. on Selected Areas in Comm.*, vol. 24, no. 11, pp. 1960 – 1971, November 2006.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7101
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7101

[6] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *MobiCom '05: Proc. of the 11th annual intl. conf. on Mobile computing and networking*. New York, NY, USA: ACM Press, 2005, pp. 73–87.

[7] Y. Ganjali and A. Keshavarzian, "Load balancing in ad hoc networks: Single-path routing vs. multi-path routing," in *INFOCOM 2004: Proc. of the 23rd annual joint conf. of the IEEE Computer and Comm. Societies*, vol. 2. IEEE, March 2004, pp. 1120–1125.

[8] S. Tai, R. Benkoczi, H. Hassanein, and S. Akl, "A performance study of splittable and unsplittable traffic allocation in wireless sensor setworks," in *2006 IEEE Intl. Conf. on Comm.*, vol. 8. IEEE, June 2006, pp. 3432–3437.

[9] A.-A. Mahmood, E. S. Elmallah, and A. Kamal, "Non-bifurcated routing in wireless multi-hop mesh networks," in *LCN '07: Proc. of the 32nd IEEE Conf. on Local Computer Networks*. IEEE, October 2007, pp. 279–286.

[10] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan, "End-to-end packet-scheduling in wireless ad-hoc networks," in *SODA '04: Proc. of the 15th annual ACM-SIAM symp. on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2004, pp. 1021–1030.

[11] A. Raniwala, K. Gopalan, and T.-c. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE Mobile Computing and Comm. Review*, vol. 8, no. 2, pp. 50–65, April 2004.

Digital Object Identifier: 10.4108/ICST.BROADNETS2009.7101
http://dx.doi.org/10.4108/ICST.BROADNETS2009.7101